FastTrak™

FastTrak and FastCard Hard Drives

User Manual

This equipment generates and uses radio frequency energy and if not installed and used properly, that is, in strict accordance with the manufacturer's instructions, may cause interference to radio and television reception. It has been type tested and found to comply with the limits of a Class B computing device in accordance with the specifications in Subpart J of Part 15 of FCC Rules, which are designed to provide reasonable protection against such interference in a residental installation. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient the receiving antenna
- Relocate the computer with respect to the receiver
- Move the computer away from the receiver
- Plug the computer into a different outlet so that computer and receiver are on different branch circuits

If necessary, the user should consult the dealer or an experienced radio/television technician for additional suggestions. The user may find the following booklet prepared by the Federal Communications Commission helpful: "How to Identify and Resolve Radio-TV Interference Problems." This booklet is available from the U.S. Government Printing Office, Washington, D.C., 20402, Stock No. 004-000-00345-4.

THIRD PRINTING December 1989

All specifications subject to change without notice.

891201

FastCard™

Fasttrak™

Hard Drives for the Amiga® 500/1000/2000

User Manual

Xetec, Inc.
2804 Arnold Rd.
Salina, Ks. 67401

Table of Contents

INTRODUCTION

- *Before You Begin*
- *Using This Manual*

Before You Begin

Thanks for choosing a Xetec hard drive for your Amiga® computer. We've designed a system of hardware and software that makes the drive as easy to use as a floppy, yet is still packed with numerous options and special features.

This User Manual covers the preparation and use of various Xetec hard drive systems for the Amiga:

FastTrak models for the A-500 and A-1000
FastTrak without drive
FastCard models for the A-2000
FastCard without drive

To keep things simple, we'll refer to all these systems by the same name in this manual: the *FastTrak* drive system. The "hard drive interface" refers to the host adaptor in *FastTrak* systems and to the interface card in *FastCard* systems. All these systems come with the same boot floppy, called the *FastTrak* boot floppy.

Using This Manual

Since your hard drive system, expectations, and level of expertise may be drastically different than other readers of this manual, the correct way for you to approach reading it may not be the best way for someone else. Use the following as a "road map" to help you know which chapters are most important to you and in which order to read them.

No matter what your situation, you should competely install your system according to the included Installation Manual before using this manual at all.

Purchased a complete hard drive system, just want to get up and running

Be sure to read **Using the Hard Drive**, p. 3, **Booting the System**, p. 19, and other chapters as desired. Ignore the chapter on **Formatting**.

Purchased a complete hard drive system, custom partitioning desired

First read **Using the Hard Drive**, p. 3, to get the drive operating. If you desire, copy off the demos and other freeware that exists only on the hard drive. Next read **Partitioning**, p. 7.

Purchased a host-adaptor or interface card only, supplying own drive

First follow the directions in **Formatting**, p. 35. Next read **Partitioning**, p. 7. Finally, read **Using the Hard Drive**, p. 3, and **Booting the System**, p. 19.

USING THE HARD DRIVE

- *Power-up/Power-down Sequence*
- *Avoiding Disaster*
- *Using the Drive with DOS*

If a hard drive was included in your drive system, it has already been prepared for you at the factory. **Do not** format or partition the drive at this time or you will lose the free software and data it contains. It has been setup to look like one huge drive to the computer. Later, you may want to split it up into several drives for convenience. For now, however, let's get the drive up and running just as it is. Besides, you will probably want to copy the software off the drive before doing anything drastic to it.

Power-up/Power-down Sequence

A computer system with a *FastTrak* drive installed should be turned on and off in a specified order to ensure proper operation and safety of the data. If all devices are plugged into a power strip, you can turn the sytem on and off with the "master switch" on the strip. If you must turn the devices on and off individually, follow these sequences.

Power-up

- All peripherals including the monitor and hard drive
- The computer itself

Power-down

- The computer
- All other peripherals

These sequences are absolutely mandatory if the host adaptor was configured to draw power from the hard drive (*FastTrak* models only). Notice that these sequences would not apply to *FastCard* users, since the drive and interface are powered up and down with the computer (except for external add-on drives). Also note that any special power sequences given for your other peripherals should still be followed.

Avoiding Disaster

Experienced Amiga users know that you have to be very careful when you want to reboot the computer or shut it off. When you finish reading or writing to a floppy, the drive continues to run for another few seconds. The computer must not be shut off or rebooted until this period is up (when the drive light goes out).

With a hard drive, you must be even more careful, because you cannot tell if the few seconds are up by just looking at the drive light; it will go off then possibly flicker a few seconds later. If you reboot the system or shut it off before this final pulse, **your data will be corrupted!** Files already safely on the drive can be affected too! This is just the way the Amiga's buffering works. This quirk usually strikes people who get in a hurry to shut off their computer or hard drive the second a file is finished saving. This simple technique will prevent such a disaster:

1) Wait for disk I/O to complete (drive light goes off)

2) Count 3 seconds off

3) Reboot (CTRL *A* A) or shut power off

This gives the DOS time to fully complete all disk I/O. Don't worry if you don't see a pulse, just wait the three seconds and you're safe. Keep in mind you only need to do this if you just finished accessing your drive.

Using the Drive with DOS

Later we will discuss different ways to customize your hard drive. For now, just turn the system on (remember the sequence shown earlier). If your host adaptor jumper was set to autoboot, the computer should begin to boot shortly. If not, you will need to insert the *FastTrak* boot disk in the floppy drive. When everything finishes, you should see a new icon for the hard drive. At this point, you can use your hard disk just like you're accustomed to using floppies. For example, you can double-click on its icon to open it, or you can use it from CLI. Its current device name is **dh0:** AmigaDOS™ accepts this as a device just as freely as it accepts other device names (like **df0:**, **par:**, **con:**, etc.). To copy a file from the hard drive to floppy, for example:

copy dh0:filename df0:

Don't worry if you feel uncomfortable using CLI. A powerful backup/restore program called X-chive is included which you can use to install your floppies on the hard drive. X-chive is covered in detail in the chapter **Data Archiving**.

Several cases in which you cannot use your drive like a large floppy:

1) In Workbench,™ you cannot drag a hard drive icon onto a floppy icon or vice versa to copy the entire disk. This is not surprising, since you are only allowed to do this between two drives of equal size.

2) Some programs written specifically for floppies (direct access programs that bypass DOS) such as sector editors or un-copy protectors may not work correctly on a hard drive because it's configured differently. The only significant loss here is the sector editor, so one is included specifically for use on the hard drive (experts only).

At this point, you should play with your drive to get a feel for its use and speed. Use it from Workbench or from a CLI window, whichever you prefer. For starters, double-click on the **Slideshow** icon in dh0: and watch the sparks fly. Also, try this from a CLI or Shell:

dir dh0: all

What you do now is up to you. You may want to try some more of the demos included on the drive. Or maybe you're anxious to try installing some of your software on the drive. Enjoy. We strongly recommend, however, that before copying stacks of floppies onto the drive, you put a little forethought into the future of your drive by reading the next chapter.

PARTITIONING

- *What is Partitioning?*
- *Advantages of Partitioning*
- *Using the* Partitions *Utility*
  - *Entering Partitions*
  - *Simple Mode*
  - *Complex Mode*
  - *Checking Partitions*
  - *Formatting Partitions*
  - *Using New or Edited Partitions*
  - *Variable Sectors per Block*
- *Partition Info Details*
- Partitions *Messages*

What is Partitioning?

A major decision that faces every owner of a new hard drive is partitioning. What? You never had to do that with your floppies? Consider what you just bought — a disk that someday could hold tens of thousands of files (depending on their size, of course). If you ever get to that point, you will find yourself frequently wading through masses of directories and sub-directories. If you're typical, you won't be able to find some long-forgotten file you are looking for, but you will find plenty of other files whose purpose you've long since forgotten. Only those who have previously used hard drives fully realize how easily junk or temporary files accumulate.

The point here is that organization is very important for a hard drive and its data. Splitting a hard drive into partitions, or **partitioning**, helps you keep files organized. So what is a partition? Each partition acts like a separate drive, each with its own device name, even though they all are contained on the same physical disk drive.

Advantages of Partitioning

As has been stated, the main reason for splitting your beloved behemoth of storage into smaller partitions is to keep its files organized. A file written in a partition is visible in that partition only. Here's an example of a good partitioning scheme: a medium-sized partition called **text:** which holds your WP files, a partition called **PD:** that holds your collection of public domain software, a large partition named **pic:** for all your picture files, and a large partition named **dh0:** that contains everything else. As you can see from this example, the drive can be customized for your use, specifying the number and size of partitions, as well as the name of each. Even if you don't want to split up the drive, it still must be partitioned (you just specify one humongous partition named whatever you want).

Another advantage of multiple partitions is speed. DOS works with files quicker when they are separated into small groups. Using partitions and sub-directories both help in this area.

One not-so-obvious advantage to partitioning the drive is data safety. Each partition is so much like its own drive that it can be formatted without disturbing the others. Dwell on this thought for a moment, if any data ever becomes corrupt, only the involved partition need be reformatted, the rest of your data is safe. You can even use a partition as an area to backup files from other partitions on the same drive. Although not 100% safe, this makes for a "quick and dirty" backup system.

One final advantage involves systems that are capable of auto-booting from the hard drive. A separate partition is usually dedicated to booting the drive, which then installs the other partitions. The reason for this ''chaining'' is speed as you'll see in the section titled **Auto-Booting** (p. 20).

Partitioning has only one disadvantage — memory used. Each partition used can consume between 13K and 28K depending on the partition. Each partition however, doesn't consume this memory until it is being used. In other words, once you attempt to use a partition, it goes and grabs its chunk of memory, which is ''non-refundable'' (until you reboot or power-down, of course).

Using the *Partitions* Utility

Caution: remember that changing a drive's partitions will make any data previously on that drive inaccessible.

Okay, so you're convinced that you should partition the drive. So let's get to it. Your first step is to *plan ahead*. You're going to have to foresee right now what kind of files and roughly how many you think you'll accumulate over the years. If you make a bad guess, and in a short time one partition is filled while the others are barely used, you *can* repartition the drive. Unfortunately, when you move partition boundaries or change handlers, the partitions must be reformatted, meaning all the data will have to be copied elsewhere temporarily (to floppies, streaming tape, or another hard drive).

Note: if you intend to auto-boot your system from the hard drive (see the section on the operating system jumper in the Installation Manual to see if your system will be capable), you must define a special boot partition in addition to all the other partitions you need. Without it, you will not be able to run your main partition at its top speed. This partition can be located anywhere on the drive, but it needs to be at least 75K in size, must use the Standard handler, and should be named *BOOT*. Drive systems that came with a hard drive have this boot partition already prepared for you. For users with multiple drives, only the boot drive needs this boot partition. Details of how to use this partition are discussed in **Auto-Booting**, p. 20.

On the *FastTrak* boot disk is a program called *Partitions*. Its sole purpose is to help you setup the partitions you want and prepare them for DOS to use. It will only work on drives that have been formatted and prepped with the *SCSItools* program, covered later. Since this was done to your drive at the factory, go ahead and start *Partitions* by double-clicking one of the icons (either ''simple'' or

''complex'') in the **utilities** drawer, or by typing ''partitions SIMPLE'' or ''partitions COMPLEX'' in CLI. The program, as is obvious, can be run in a simple or complex mode, depending on your expertise. The simple mode is fairly foolproof for the beginner; it does most everything automatically that is done manually in the complex version. Setting things up manually, if you feel capable, gives you much more freedom to setup the drive precisely as you want it, as well as giving access to some features that are just not possible from the simple mode. Note that you can setup the drive in the simple mode, exit the program, and restart it again in the complex mode to look things over, make changes, etc.

After a few moments, the window in fig. 1 should appear. Let's look briefly at several features of this window. The small box in the extreme upper-left is the *close* gadget. Use it to exit the program. **Never exit the program by shutting the computer off or rebooting!** The program will save its final data as you exit. The four boxes in the upper-left are your commands. Clicking on these with the mouse allows you to perform the various functions of *Partitions*. The box in the upper-right is a text window where the program can 'talk' back to you, giving error messages and helpful info about what you're doing. Notice that at this time the ''harddisk.device'' revision is displayed as well as the mode in which you selected *Partitions* to run. The 14 boxes marked **Drive units** represent the physical hard disk drives that can be attached to your *FastTrak* hard drive interface. For every drive present, its name will appear in the corresponding box. The eight boxes running across the bottom are where you will enter the names of your partitions (up to 8 per drive).

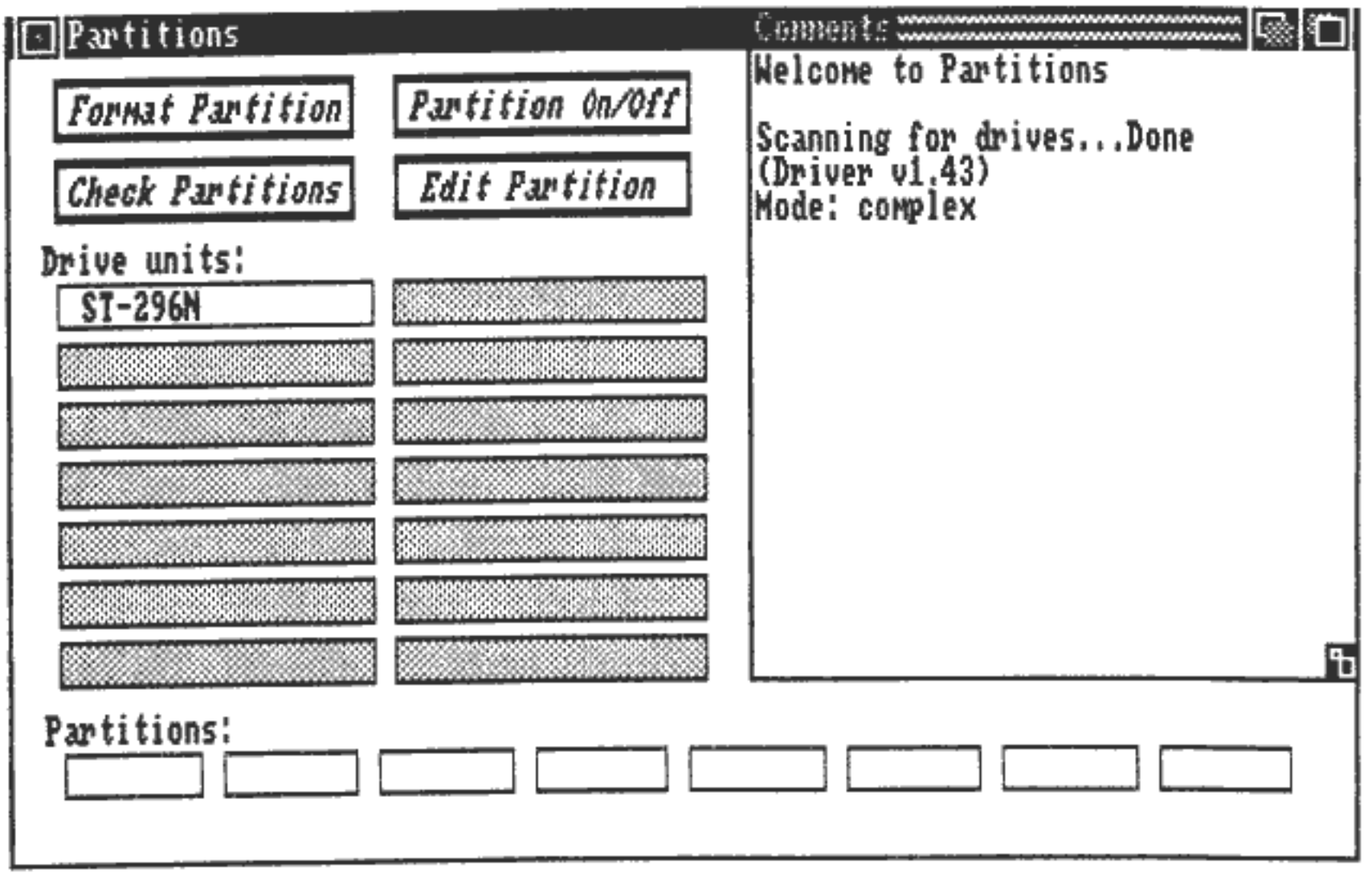

Fig. 1 – *Partitions* Main Screen

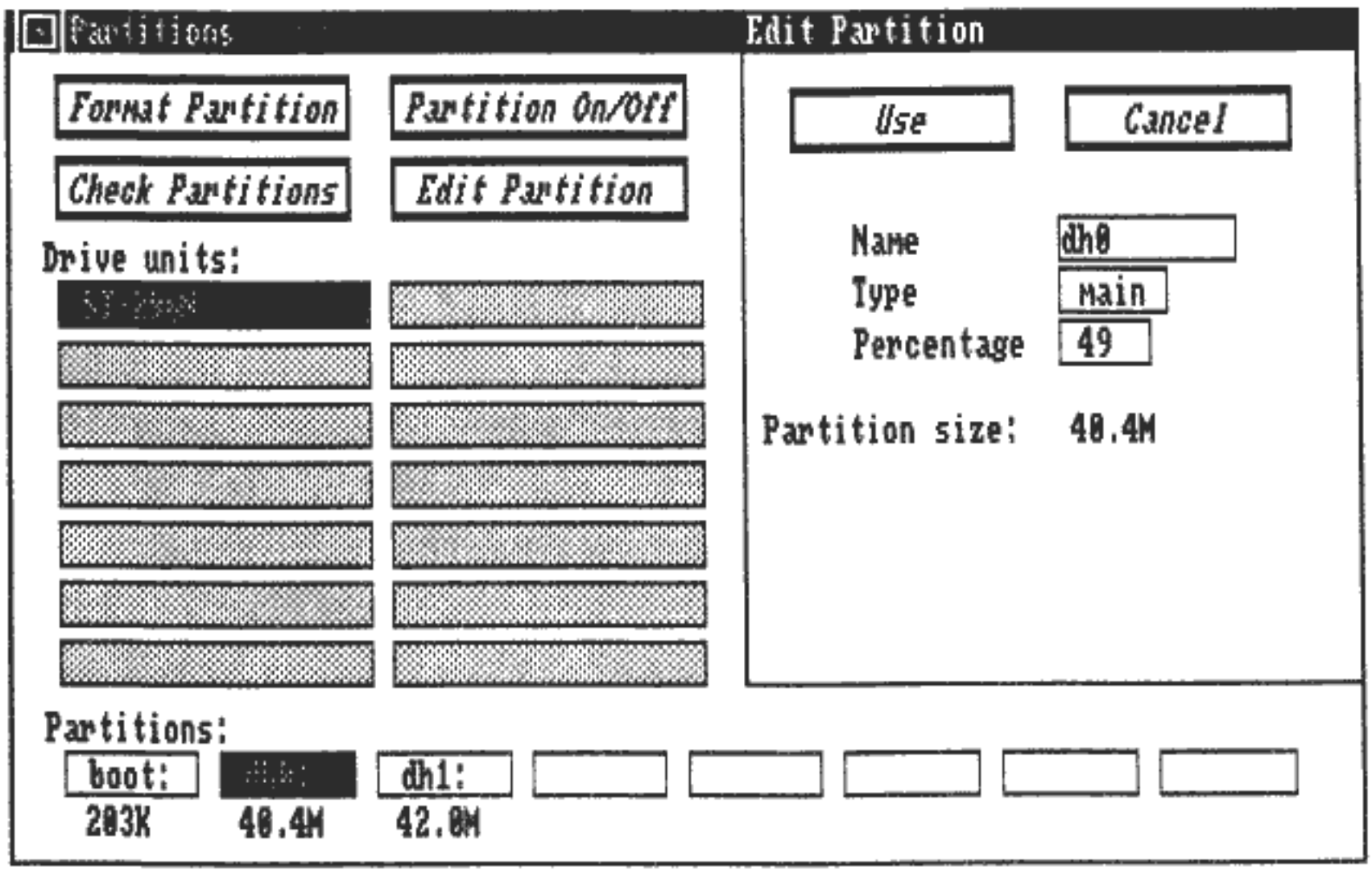

Fig. 2a – *Partitions* Simple Editing Screen

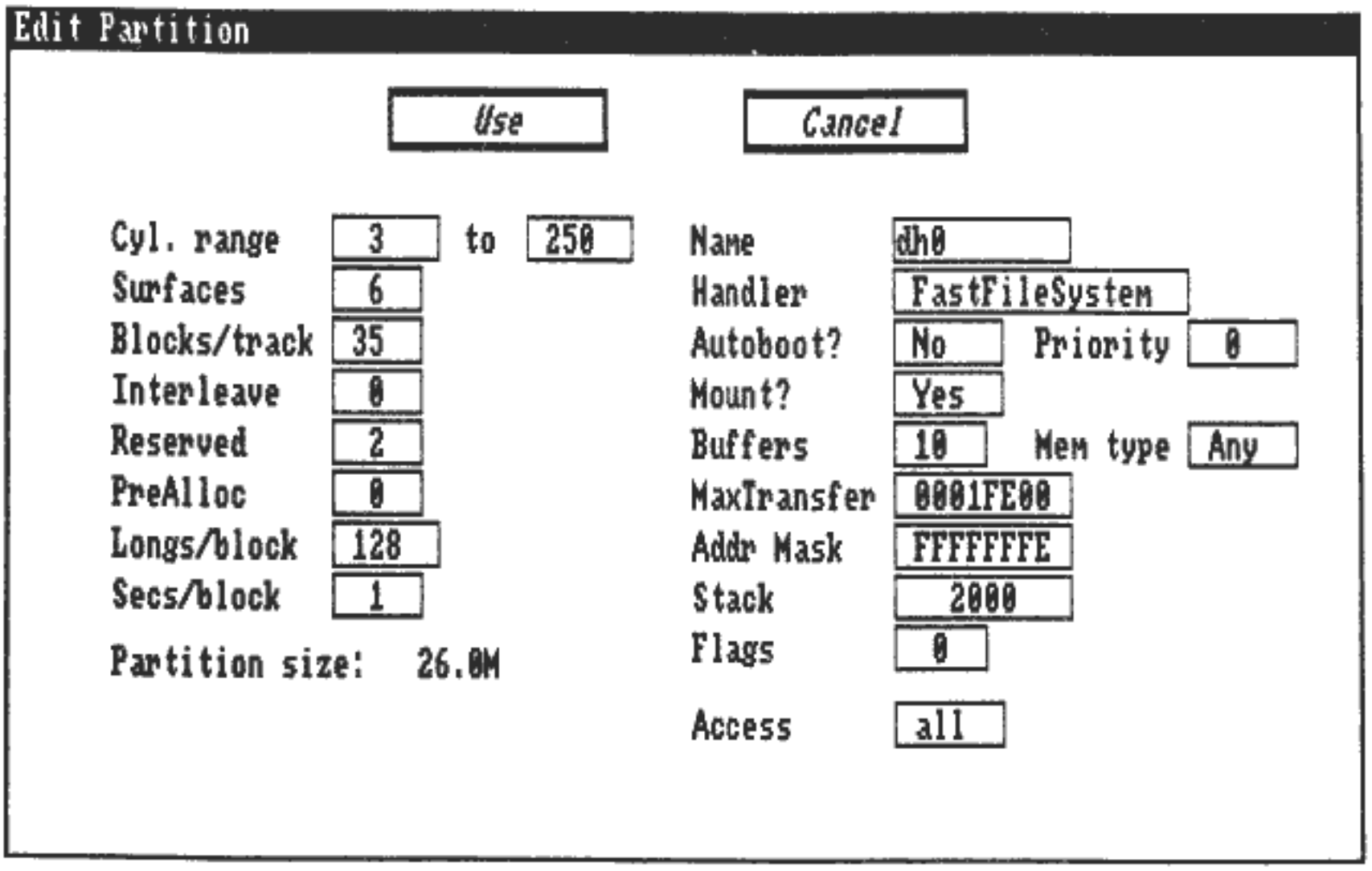

Fig. 2b – *Partitions* Complex Editing Screen

Entering Partitions

The first step to partitioning your drive is to enter the partitions, of course. Referring to fig. 1, click on your drive in the **Drive units** section. Its name should now be highlighted, meaning that's the drive you will be working with. If you

have more than one drive connected, you will notice that only one can be highlighted at a time. Since each drive is a separate entity (with its own name, size, parameters, and partitions) you must work with one drive at a time.

Once your drive is highlighted, **dh0:** (and possibly **BOOT:**) should appear in a partition box (if the drive was setup at the factory). That is the name of the partition setup to fill the entire drive. You can edit or erase this partition. For now, let's leave it alone and just create another one. To do this, click on any of the empty partition boxes (their position is meaningless) then click on the **Partition On/Off** command box. You just created a partition whose name is "???" (you still need to edit it).

Click on **Edit Partition** to begin editing your new partition. The screen shown in fig. 2a or 2b should now appear (depending on the mode: simple or complex). This screen contains all the user-changeable information for a partition. To edit any of the data, click on its box and use the **DEL**, **BACKSPACE**, and cursor keys. You can clear the data by holding Ⓐ and pressing **X**. Let's look at each mode's screen separately.

Simple Mode

Editing a partition in this mode is simple. Enter the name of the partition in the name box, click in the type box until the correct partition type is displayed ("main" for all but the boot partition), and enter in the bottom box the percentage of the total drive to be used for this partition. When you press RETURN, the drive's size will be updated below. If you enter a percentage that is too high (based on other existing partitions), it will be lowered to the maximum value possible.

Complex Mode

Only a few of the items need to be discussed right away. For the others, see p. 16.

The data on the left side of the editing screen describes the size of the partition and the geometry of the drive that contains it. Most of the data is correctly set for you (this information is inherited from *SCSItools*). You will normally be concerned only with the two boxes marked **Cyl. range**. A **cylinder** is a unit of storage on your drive. By specifying the cylinder range, you confine each partition to its own share of the hard drive. The size of this range directly affects the partition's maximum storage. Each cylinder of a hard drive must be contained in the cylinder range of only one partition. Another way of saying this is that the range of each partition should just meet, but not overlap. Some examples will clarify; both of these are sets of properly defined partitions:

1 — 199, 200 — 453, 454 — 602

124 — 189, 426 — 611, 1 — 123, 190 — 425

All of these examples are improper:

1 — 200, 200 — 454, 454 — 602

1 — 602, 200 — 300

1 — 299, 605 — 300

Notice the last example has its second partition's range specified backwards. This is unacceptable. A newly-created partition's cylinder range will default to the maximum range allowed on the drive. You should never try to exceed these initial values.

The data on the right side of the editing screen tells DOS how to treat this partition. The **Name** entered here is the device name DOS will associate with this partition. Notice something; since each partition has a unique *device* name, your computer thinks that each partition is actually a drive of its own — here again, the beauty of partitioning. Other system device names are usually three letters long (df0, par, con, etc.) but you can make yours longer if you want. (DOS likes device names best that are three or four characters long.)

The **Handler** is a multiple-choice item that changes each time you click on it. This specifies the type of file-handler that DOS should employ to talk to this partition. The standard handler is the old standby that is used for floppies and hard drives up to now. FastFileSystem is a new handler that works much quicker with hard drives. As you will see in the section on auto-booting (p. 20), any partitions that are to be mounted at boot-time can use only the standard handler.

To exit the editing window, you need to click either **Use** or **Cancel**. *Use* will save the current settings, where *Cancel* will not. Both return you to the main *Partitions* screen.

Go ahead and create/edit as many partitions as you wish. Remember that you must highlight the partition you wish to edit before clicking *Edit Partition*. Notice that the current storage capacity of each partition is listed below each partition box in the main screen.

Checking Partitions

Once you are satisfied with all the partitions for your drive, you should select the command box labeled **Check Partitions**. A screen similar to fig. 3 will appear, listing the name of the drive being checked and each partition defined for it. Each partition will be analyzed, and any errors or questionable parameters will be called to your attention. You have the freedom to ignore them, but we strongly discourage using any partitions that don't check out as "okay." If many errors are found, you can pause the scrolling to read them by holding down the right mouse button.

Check Partitions only tests the partitions for the current drive unit, whose name is currently highlighted. If you have more than one drive, for each you must highlight it and select *Check Partitions*.

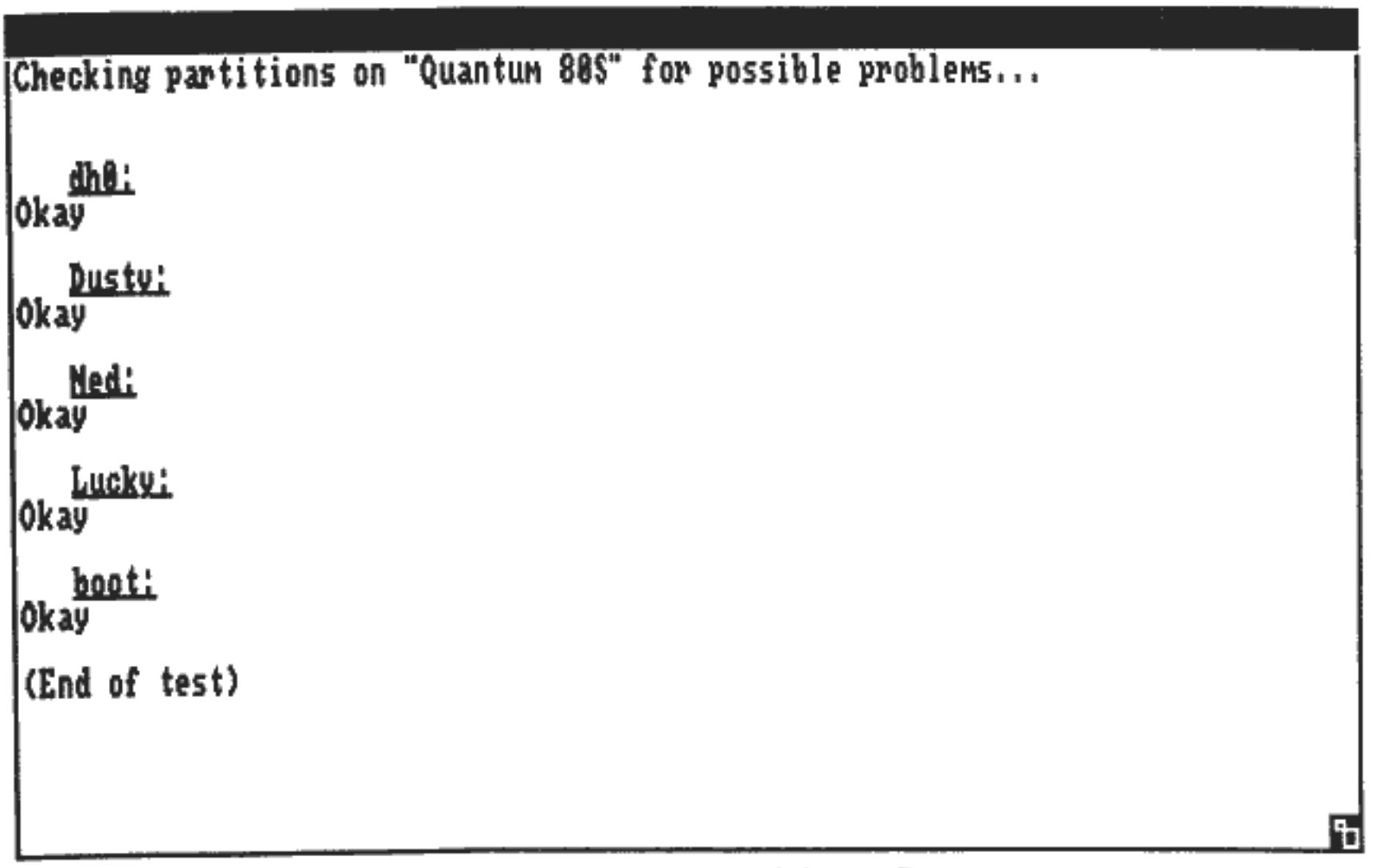

Fig. 3 – Check Partitions Screen

Formatting Partitions

Before a partition can be used by DOS, it must be formatted. But the drive already is formatted. What's the deal? Well, for your hard drive, there is a **low-level** and **high-level** format procedure. The low-level format, which is performed by the *SCSItools* program, prepares the magnetic surfaces in the entire hard drive so that they can store data as well as mapping out defective spots. This procedure is already done for you. A high-level format only affects one specific partition, erasing any files and preparing it for use by AmigaDOS™. A high-level format is performed if you select an icon and choose *Initialize*, if you use the CLI *Format* command, or if you select the "Format Partition" command in *Partitions*. If you try to use a partition without high-level formatting it, you will get a message saying **Not a DOS disk** or **Error Validating Disk**. Fig. 4 shows graphically what each type of format does.

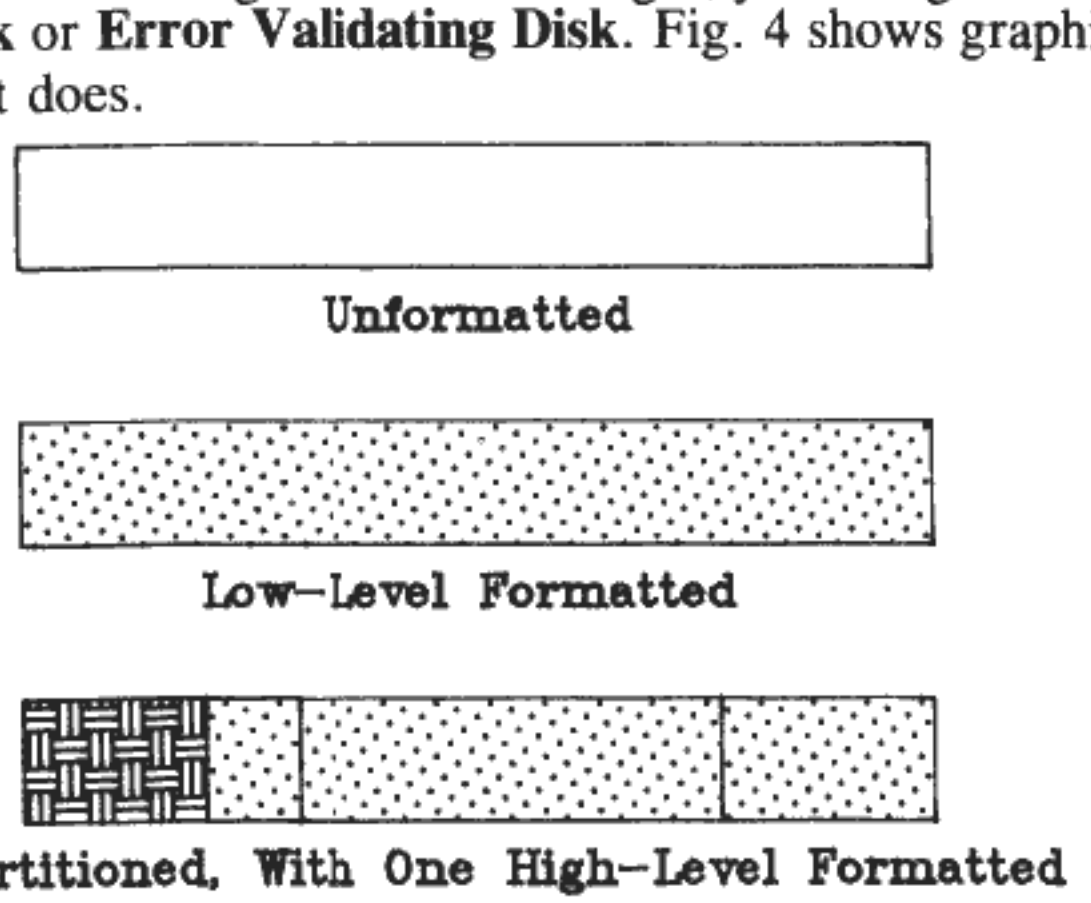

Fig. 4 – Low vs. High-Level Formatting

New partitions or partitions which have been modified (using the editing screen) must be high-level formatted and the computer rebooted before they will be recognized as usable storage.

The easiest way to format a partition is from the *Partitions* program itself. After you have done "Check Partitions" to be sure everything's okay, highlight the desired drive unit, and then highlight the appropriate partition at the bottom of the screen. Next click on **Format Partition**. A requester will appear, reminding you what you are about to do. Select *Go ahead* to proceed or *NO!!!* to abort. If you choose to continue, another requester will appear, asking whether to use the quick or long method. Both methods write the data that DOS requires, logically erasing all files on that partition. The long method also verifies that every block in the partition is usable — slow, but a good idea if you want to be absolutely sure the partition is going to be okay. The quick method, however, is sufficient. Partitions formatted with the *Partitions* program will be labeled the same as their device name. In other words a partition whose device name is "Text:" will have "Text" as its volume name as well.

If you just need to re-format a partition, but you haven't changed its boundaries, it may be simpler to use the normal formatting procedure (*Format* command from CLI or *Initialize* from Workbench™). See your computer manuals for more information. You cannot use these methods to format a partition which you just created or edited (unless you reboot first).

Using New or Edited Partitions

It is very important to keep in mind that your computer's DOS acquires information about all partitions when it first boots up. That's the only time. Therefore, an edited or new partition cannot be accessed by DOS until the next time it's booted up. The reason the *Partitions* program can format one of these "mystery" partitions is rather complex, but it basically involves talking to the drive behind DOS'es back.

Variable Sectors per Block

Some drives (Quantum, for example) have a varying number of sectors per track, depending on which track you're talking about. Drives like this read "0" for the Sectors/Track entry in *SCSItools*. This is an accepted standard value meaning "variable." You cannot, however, specify 0 Blocks/Track in *Partitions*; this is the only time two values in *Partitions* and *SCSItools* should differ.

When repartitioning a drive that uses variable sectors per track, use the existing partition definitions as a template for how to do your own: notice the *Blocks/Track* entry and the minumum and maximum cylinders used by the group of partitions. When you edit them or make new partitions, use the same entry for *Blocks/Track* and don't exceed the initial overall cylinder range.

The maximum cylinder range on such a drive doesn't correspond to the "real" range specified by the drive manufacturer but is altered so that a fixed number of blocks per track can be assumed by AmigaDOS. Here's how to construct an acceptible partition from scratch on a drive with variable sectors per track. First, enter an arbitrary number for the *Blocks/Track* entry (like 20). **You must use this same value for all partitions on that drive!** Next, consult the drive's specifications to find the total number of data blocks. Divide this number by the number of surfaces (or heads). Then divide that result by the number you entered for *Blocks/Track* and subtract 1. This is the maximum cylinder number that this drive's partitions can use.

Partition Info Details

Following is a reference section for each of the partition data in the complex partition editing screen (fig. 2b).

Cyl. range: the bottom and top cylinder that this partition can use on its hard drive.

Surfaces: the number of magnetic surfaces or heads in the hard drive that this partition is on. This number should always match the number specified in *SCSItools*. It defaults to that number.

Blocks/track: the number of data blocks that can be placed on a track for this hard drive. A track is the area of disk covered by one head in one revolution. This number should match the number specified in *SCSItools* unless it is 0. It defaults to that number (except for an OMTI 3520 controller, in which case it defaults to one less).

Interleave: currently unused. Set to 0.

Reserved: number of data blocks at the bottom of the partition which should not be used for data storage. No useful purpose anymore.

PreAlloc: number of data blocks at the end of the partition which should not be used for data storage. Used by FastFileSystem only. Set to 0.

Longs/block: size of each data block in the partition, specified in longwords (4 bytes each). This should be 128 ($\times 4 = 512$ bytes/block).

Secs/block: number of data sectors per data blocks on the hard drive. Currently unused. Set to 1.

Name: device name to be used by DOS. Should not have a colon (:) in it, although when used in DOS, it must be followed by a colon (i.e. CON:).

Handler: type of program used to interface DOS to the hard disk device driver. Determines format of data and the access speed of the partition. Requests for a handler other than *Standard* will be ignored if the Autoboot flag below is set to "YES."

Autoboot?: attempt to install partition at boot time? If *NO*, will be installed at first *BindDrivers* command. If *YES*, will be installed before booting system. If set to *YES* on a system which will not autoboot, the partition will not get installed.

Priority: the autoboot priority. This is only meaningful if *Autoboot?* is set to *YES*. If system supports autobooting, and if no bootable floppy is in the drive, the bootable partition with the highest autoboot priority is used to boot the system. Never set this value higher than 4 or you will never be able to boot from a floppy.

Mount: again, a *YES* or *NO* question. Partitions set to *NO* never get installed. This is just a convenient way to disable a partition without completely deleting all its parameters.

Buffers: number of buffers to be used by this partition's handler for I/O buffering. More buffers will give a slight increase in speed at the price of 512 bytes of user memory per buffer.

Mem type: type of memory to be used for the handler I/O buffers above. Can be *ANY*, *CHIP*, or *FAST*. If you demand *FAST* on a 512K computer (which has no fast RAM), the handler will abort with no buffers; therefore the partition will not be installed.

MaxTransfer: for FastFileSystem handler only. Specifies largest single I/O request that can be passed to the device driver in bytes. Given in hexadecimal. Set to 1FE00, or if using driver version 1.40 or higher, you can use 1FFFE00.

Addr Mask: for FastFileSystem handler only. Limits address area that is used to exchange data between DOS and device driver. Each '0' bit demands a '0' in the corresponding address bit. Normally set to FFFFFFFE to accept any word-aligned address area.

Stack: size of stack (in bytes) to be used for handler. To prevent possible crashes, should be at least 2000 bytes. Different handlers will have varying safe values for the stack size. Rule of thumb: if unsure, make it bigger!

Flags: value to pass to device driver for 'Flags' in OpenDevice command. Must be 0.

Access: assigns the partition to a specific user in a multi-user scsi network (or ALL). If the drive has a single computer connected, set this to ALL. See **SCSI Networks**, p. 45, for more information.

Partitions Messages

The following are various messages that can occur in the feedback window or *Check Partitions* window of the *Partitions* program:

A boot priority of *n* will allow no floppy boot: the priority specified is so high that autobooting from the hard drive will always override a bootable floppy.

Autoboot partitions must use standard handler: the autoboot flag is set to *YES* while the handler is something other than *Standard*.

Can't format, will destroy Cyl 0: a format will always abort if the partition's cylinder range contains cylinder 0. It must never be disturbed, since important drive information is kept here for the *FastTrak* system.

Cylinder conflict with partition "*nnn*": this partition's cylinder range overlaps in some way with the cylinder range of the partition listed.

Cylinder range backwards: the low end of the cylinder range is a larger number than the high end number.

Cylinders above limit for drive (*n*): the partition's cylinder range goes higher than the highest usable data cylinder on the hard drive. The limit is shown in parenthesis — use it!

Cylinders below limit for drive (*n*): the partition's cylinder range goes lower than the lowest usable data cylinder on the hard drive (usually 1). The lowest cylinder is shown in parenthesis.

Cyl. range extends too high!: *Partitions* tried and failed to access the last few data blocks in the partition as it is defined.

Drive was setup with *n* blocks/track: the blocks/track entry for this partition differs from the same entry given in *SCSItools* for this hard drive.

Drive was setup with *n* surfaces: the surfaces entry for this partition differs from the same entry given in *SCSItools* for this hard drive.

Fatal error, harddisk.device not found: *Partitions* cannot even begin to operate because of the absense of this important system program. This file is only available when the system is booted from a disk that contains the Xetec icon and *harddisk.device* both in the *Expansion* drawer. This can also occur if the host adaptor didn't config into the system.

'Flags' = *n* (shame, shame): the partition's *Flags* entry should never be non-zero.

Must have some buffers: handler will not operate without at least one buffer.

No info on unit *n*: drive information created by *SCSItools* is absent from this drive. *Partitions* cannot be used on such drives until adequately prepared in *SCSItools*.

Nonstandard block size: *n* longwords: the Longs/block entry is not the standard value (128).

Nonstandard sectors/block: the secs/block entry is not the standard value (1).

(Unit *n*): a new drive unit icon was highlighted; this is the unit number it is known as to the system.

Wimpy stack: The stack size specified seems dangerously low.

BOOTING THE SYSTEM

- *Auto-Booting*
  - *Attaining Maximum Speed*
  - *Making an Auto-Booting System*
  - *Optimal Unit Numbering*
- *Quick-Booting*
- *Changing Preferences*
- *Startup Scripts*

The boot disk included with your *FastTrak* system is intended to be used for booting your computer and hard drive (be sure to make a backup copy!). Unfortunately, due to all the hard drive utilities included on the disk, some standard Workbench™ files have been omitted.

This chapter details how to setup a fully auto-booting system (v1.3 only) or at least a quick-boot floppy (v1.2). Both methods of booting will give you access to all Amiga system files and programs, as well as the special programs present on the *FastTrak* boot disk.

Auto-Booting

The *FastTrak* and *FastCard* hard drive systems have a wonderful new feature that makes it possible to auto-boot from the hard drive. What's that mean? It means your computer can startup using the hard drive instead of a Workbench™ type floppy disk. This is much more convenient, not to mention astoundingly faster. Unfortunately, not all Amiga systems will currently do this. Only computers with a v1.3 or higher Kickstart™ ROM (or disk for the 1000) will support auto-booting. See the Xetec Installation Manual for info on determining your ROM version and setting the appropriate jumpers on the hard drive interface.

For those of you who have systems that can't autoboot, the next best thing is a quick-boot floppy. See below to make such a disk.

Systems that can auto-boot, when turned on or rebooted, will first attempt to find a floppy in unit 0 (df0:). If the drive is empty, or the floppy in the drive does not have the boot blocks *Install*ed, the system will next look for all hard drive partitions with their *auto boot?* parameter set to *YES*. Of these, the one with the highest auto-boot priority will be used for booting. If no such auto-bootable partitions exist, the familiar hand will appear on the screen, beckoning you to place a bootable disk in the floppy.

The hard drive partition selected for auto-booting must have many of the same files present on a boot floppy in order for your system to come up properly. The whole process is just much faster when working with a hard drive.

Attaining Maximum Speed

You might think that the best layout would be something like the following: set aside a healthy-sized partition that will be used for auto-booting. Have it contain all system files (fonts, printer drivers, C, System, and Utilities sub-directories, etc.). This would be great, except for one system limitation. Any partition that is installed early enough to be used for auto-booting cannot use anything but the

Standard handler. What this means is that any data in that partition will be accessed very slowly compared to data in other partitions. So you don't want to cram all those system files (which are accessed so often) into such a slow place.

We recommend a chaining technique to get the most speed out of your drive. Here's how it works. You set aside a very small partition whose only purpose is to begin the boot process. This so-called **boot partition** is setup as auto-bootable in the *Partitions* program. The first thing it does when booting is to install all other partitions on every drive connected to the computer. It then passes the boot responsibility on to one of those partitions which does the rest. What does this accomplish? The small boot partition must use the slow (*Standard*) handler, but the one it passes control to can use one of the fast handlers (like *FastFileSystem*). The bottom line is that most of the boot procedure can run at maximum speed, as can all hard disk I/O once the system is up and running.

For this procedure to work, the ''handoff'' partition (the 2nd one in the boot chain) should contain all the system drawers and sub-directories (such as libs, devs, c, s, t, l, system, utilities, expansion, etc.). The boot partition, being so small, must contain only those files necessary to install the hard drive and pass control on to the ''handoff'' partition.

Making an Autobooting System

Hard drives shipped in *FastTrak* or *FastCard* systems have their boot partition already setup and contain the necessary files; all you must do to make this partition appear and function is to set your interface jumper for 1.3 (if that is true!).

If you supplied your own hard drive or in some way destroyed the data that was shipped on the included drive, there are several things you must do to make your drive autoboot.

1) First, we reiterate that you must be using a 1.3 Kickstart. If not, see the next section on quick-booting.

2) Using the *Partitions* program, make a partition that will hold at least 75K of data. Use the *Standard* handler and set *Autoboot?* to ''YES.'' Format this partition (from *Partitions*).

3) Make sure the main (''handoff'') partition is formatted and reboot the computer using the *FastTrak* boot floppy.

4) Run the utility **MakeAutoBoot** by typing its name in a CLI or double-clicking its icon in the utilities drawer. To use this program you must enter the names of the boot and main partitions in their appropriate boxes (follow the instructions on screen). Clicking on the **Install** box will write the necessary files to your boot partition, as well as creating an appropriate startup-script. Clicking on **Copy** will copy the entire boot floppy contents to the main partition.

If you want to make customizations to the startup-sequence, edit the one in your main (''handoff'') partition, *not the one in the boot partition!* See below for details.

Optimal Unit Numbering

Because drives are scanned numerically during the computer's auto-boot phase, you should have a device present at unit 0 on your SCSI bus. In other words, if you have just one drive, set it to unit 0. If you have three, set their unit numbers to anything you want, as long as one of them is 0. Having a unit present at device 0 will prevent a 30 second timeout delay when auto-booting (quite annoying).

Users with multiple drives should additionally ensure that the drive that takes the longest to initialize (from power-on) is placed at unit 0. This will ensure that all drives are properly installed, while still achieving the fastest possible autoboot.

Don't worry about changing the unit jumpers for a drive, even if it is already partitioned and contains data. The partitions and data will still be good at the new unit number.

Quick–Booting

For those with systems not capable of auto-booting, there is still a way to boot quickly. All you must do is make a **quick-boot floppy** that gets the hard drive up and running then passes control to it. Then the bulk of the startup procedures can be run from the hard drive. Here's how to make a quick-boot floppy:

1) Boot your system using the *FastTrak* boot disk (just this once).

2) Type **MakeQuickBoot** or double-click its icon in the utilities drawer. Follow all instructions on the screen. This will erase any data on the floppy that you provide.

If you want to make customizations to the startup-sequence, edit the one on the hard drive, *not the one on the quick-boot floppy.* See below for details.

Changing Preferences

Making changes in *Preferences* can be a little tricky for systems that quick-boot or auto-boot. Here's the catch-22: your chosen system preferences are read from the *system-configuration* file at boot up time. On an auto-booting system, this file must be present in your boot partition; for quick-boot systems, it must be

present on the quick-boot floppy. No problem so far, that file was copied when the initial boot partition or quick-boot floppy was made. But when you use the *Preferences* program to make changes and select *Save*, the new file is written to the SYS: drive, which is now the main hard drive partition. In other words the changes you make will not "take" unless the file is written back to the boot partition or quick-boot floppy (whichever is applicable).

The procedure of copying the new file to the correct place is simple; any time you make changes with *Preferences*, just double click the **PrefsToBoot** icon. If your boot partition or quick-boot floppy was named something other than "Boot," *PrefsToBoot* will not work. To rectify this, use *Relabel* to change its name to "Boot".

Startup Scripts

The most common customization made to Amiga systems is to the startup-sequence (or script). Everyone has their own preference of what devices to install, assigns, paths, resident programs, etc.

Your hard drive has the same startup script found on your Workbench 1.3 distribution disk. Since you are likely to want to doctor this up, let's cover a few guidelines concerning custom startup scripts:

1) As has been stated earlier, make any modifications to the startup-sequence found on the "handoff" (main) hard drive partition, not to the small one found in the boot partition (or boot floppy, if quick-booting).

2) If you use Workbench heavily (as opposed to CLI), add lines in your startup that "touch" each partition you have allocated. You can CD to them one by one, include them in a PATH command, or whatever. By using each partition in some way, you request that each be mounted immediately, thereby causing their icons to immediately appear in Workbench. This delayed-mounting was added to AmigaDOS 1.3 to conserve memory.

3) Use ED or some other text editor to make the actual modifications to *startup-sequence*.

DATA ARCHIVING

- *Familiarization*
- *Using the File Area*
- *Backing-Up Files*
- *Incremental Backups*
- *Restoring Files*
- *Verifying Files*
- *Archiving Large Files*
- *The Archive Bit*

A very important part of owning a hard drive is archiving, or backing up, the data it contains. Hard drives are designed to be as trouble-free and foolproof as possible, but the possibility of mistakes, power failures, sabotage, etc. is just too great to risk your data. Our goal is to make archiving so easy and quick that it becomes a habit. Xetec offers the **FastTape** series of streaming tape systems, a companion product to the *FastTrak* that will backup or restore your data at the rate of up to 5 megabytes per minute using a removable streaming-tape cassette. For those not able to justify its cost, a powerful floppy-oriented archive utility called *X-chive* is included in the utilities drawer. (Note: technically, the program is pronounced Zee-Kive, since the "X" stands for Xetec.)

Familiarization

Start the *X-chive* utility by double-clicking its icon or typing X-chive in a CLI or Shell. A screen should appear that looks like fig. 5. Spend a few moments to become familiar with its various features.

Fig. 5 — X-chive main screen

The large black area is called the file area. Directories can be displayed here. Below the file area are four command buttons that manipulate the file area in different ways. In the upper-right are images of a hard drive and of a floppy. The arrow between them will start a backup when clicked on. Later, you can make the restore and verify arrows appear. Below this is the floppy info box.

This shows information for the floppy currently in each drive: the number of free data blocks on the floppy, the percentage of free area used on the floppy, and its status. If a floppy drive doesn't exist or has nothing in it, the info for that drive will be blank. In the lower-right corner is the message box. Updates on what's happening and error messages are all printed in this scrolling text area.

Using the File Area

Before trying to do a backup or restore, let's get comfortable with using the file area and associated commands. First, call up the directory of one of the devices listed in the file area (how about a hard drive partition) by clicking on it to highlight it, and then clicking on **Get Dir**. Another way to do this is to double-click on the device name. After a short pause, the directory should be displayed. If it's longer than the file area box, you can use the scroll gadget just to the left of the box to look through it.

Notice the **Path** box just above the file area. It will always show what it is that's displayed in the file area (which device and what directory). You can type a path directly into this box if you prefer. It will be displayed as soon as you press *RETURN*.

Also, notice **Volume:** just above it. This displays the volume name of the device whose directory is currently displayed (useful when the directory is from a floppy).

Now double-click on a sub-directory in the file area (or highlight it and click on **Get Dir**) to display it. Notice the change in the **Path** box. Now click on **Parent**. This is the directory that you were just looking at before. Click **Parent** again and the list of current devices will be displayed. This is all the higher you can move. You can proceed directly to this device list by clicking **Devs**.

Backing–Up Files

Each partition, acting like a separate drive, is backed up separately. You may even backup a more specific group of files, such as a certain sub-directory or even one specific file.

Let's try backing up some files. Use the file area to call up some directory. Now highlight all the files or directories you want backed-up (click on **Toggle** to highlight them all). Now click on the backup arrow (points to the floppy). An options window will now appear. First, select which floppy to direct the backup to (only ''real'' drives that have a floppy in them will be accepted).

Next you need to make your choice for the three options. For each, click in the box to display the next choice. *Files* (all, new) selects whether to copy all files or just those which haven't been backed-up yet (checks the archive bit). *Subdirs* (on, off) selects how to handle sub-directories. If *subdirs* is off, just the files in sub-directories will be copied; if on, the sub-directory's files and sub-directories will be copied. *Archive* (set, no) selects whether to set the archive bit for the files backed-up (use ''yes'' for normal handling of the archive bit).

Click on **No!!!** to abort or **Do it** to proceed.

Next will appear the floppy action window. This window gives you the option of quitting (**QUIT**), starting the backup (**OKAY**), or formatting the floppy (**EMPTY** or **FORMAT**). **FORMAT** does a full disk format, whereas **EMPTY** just re-formats it quickly. For both, enter the volume name of the disk in the box before you click to start the format.

Finally, the backup now begins. As each file is being copied, its name is printed in the message window. Notice that the info for the floppy is constantly updated in the floppy info area as the backup proceeds.

If the disk becomes full, the floppy action window will re-appear. You can either insert a blank floppy and click **OKAY**, or you can format another one.

You can abort the backup while it's in progress by clicking once again on the backup arrow. After the current file is copied, the backup will abort.

Incremental Backups

A very good habit to form is to make frequent *incremental* backups of your hard drive partitions. An incremental backup only copies files that have been created or changed since the last backup. This type of backup is quicker and requires fewer floppies than a complete backup. To do an incremental backup, just set the *Files* option to ''New.'' (Remember, the first backup should use the ''all'' option.)

Since incremental backups only copy the new files, you must keep on hand all the sets of backups made. Only when you do a complete (''all'') backup should you dispose of older backups for that partition/directory.

Restoring Files

Hopefully, you will never have a hard drive catastrophe. If so, or if you need access to older versions of files which have been revised, you will want to use the restore function of *X-chive*.

The restore process is very similar to backing up. First of all, you need to select the files to restore, so get a directory of the floppy by clicking **Devs**, highlighting the desired floppy drive (such as df0:), and clicking **Get Dir**. The backup arrow now turns into restore and verify arrows. Highlight all the desired files and/or directories and click on the restore arrow.

The restore options window will now appear. First, in the **File destination** box enter the destination path to which you want the files restored. You must at least enter a device name. Notice that you can specify a place other than from where they originally came! You can restore files to the ram disk, for example, to just gain temporary access to them. Keep in mind that you can always directly access the files on the floppies (they use the standard AmigaDOS format).

Next set the three options. They work just like they did for backups. Click **Do it** to start the restore. Each file is listed in the message window as it's being copied. Click on the restore arrow once again if you ever decide to abort the restore.

To completely restore a hard drive from multiple (incremental) backups, restore the newest floppy first (set *Files* to ''new''). Repeat this process for all remaining incremental backup floppies, going in reverse chronological order.

Verifying Files

To verify files, follow the directions given for restoring (except click on the verify arrow, of course). Instead of copying files from the floppy to the given destination, files on the floppy are compared with those at the **Verify source**. Each file's name is printed as it's being verified, and the result of the verify is also shown. The exact same information is also printed in an AmigaDOS window called *X-chive*. In this window you can more easily read the results of numerous verifications. In this window, use the mouse menu button to pause the process while you read it or make notes.

Click once again on the verify arrow if you ever want to abort the process.

Archiving Large Files

X-chive will backup and restore files too large to fit on one floppy. To accomplish this, it chops the files into segments that are put on separate floppies. A ''chopping file'' message during backup says that this is going on. To restore these fragments back to their original file, you must first restore all the fragments to the same directory; they will look like this: *original@@A, original@@B,. . . original@@!*

where *original* was the name of the huge file. To then convert these back to a solitary file, highlight any one of the fragments and select *Special/Merge files* from the menu.

The Archive Bit

Every file on your floppies or hard drive has a special flag called the **archive bit**. Its purpose is to flag whether or not that file is currently backed-up. The color of a file's name in *X-chive*'s file area shows the status of this bit: white means it's archived (the bit is set), orange means it's not archived (the bit is clear). You can also view this flag with the **list** command, which will display something like

```
SomeFile        867 ---arwed Today    09:25:53
AnotherFile     105 ----rwed Today    11:17:29
```

In this example, *SomeFile* is backed up, but *AnotherFile* is not.

The archive bit is instrumental in doing incremental backups. When backing-up files with *X-chive*, as long as the *Files* option is set to "new," only the files in the source partition/directory whose archive bits are clear (not archived) will be copied.

X-chive normally sets the archive bit of each file it copies (if *Archive* is "set"). The way a file's archive bit usually becomes cleared is when you modify or rewrite the file. Also, when you create a new file, its archive bit is not set (because it hasn't yet been backed-up).

You can directly set or clear the archive bits for specific files. This is not normally necessary as AmigaDOS and *X-chive* both automatically change the bits as necessary. If, however, you do ever want to tamper with them, do it like this: use the file area to display and highlight the file or files whose bits you want changed. Then select **Archived** or **Not archived** under the menu heading **File Status**.

A good example of needing to directly change any archive bits is when you want to "fool" *X-chive*. For example, you might want to mark the entire C directory as archived so that it isn't backed up (because you already have copies of the programs on your Workbench disk).

OTHER INCLUDED SOFTWARE

- *Park*
- *PrefsToBoot*
- *MakeAutoBoot*
- *MakeQuickBoot*
- *XetecFormat*
- *BoardList*
- *DeviceList*
- *Revisions*
- *SectorEdit*
- *TouchAll*

This section serves as a reference to various miscellaneous programs that can be found on the *FastTrak* boot floppy. Heed the cautions for each, as some of them can be very dangerous in the hands of novices.

Other software and demos can be found on the hard drive when it's shipped from the factory. This software is all public domain and is included as available. No guarantee is made by Xetec as to which programs are included and their reliability.

Park

This command resides in the utilities drawer of the *FastTrak* boot floppy. Used from Workbench or CLI, it parks the heads of your hard drive(s). ''Parking the heads'' means to move the read/write heads in the drive to the safest spot for themselves and the magnetic media. Some drives even stop spinning when parked.

From Workbench, simply double-click on the park gadget to park all hard drives. From CLI, you list the drive's unit number as a parameter after the park command. If no parameter is given, all drives connected to the system will be parked. For example to park just unit 0:

park 0

When the park command completes, a box will appear in the upper left of your screen that says ''unpark''. If you parked the drive by mistake, you may restart them by clicking on this.

Be aware that data is not accessible on a parked drive.

PrefsToBoot

This program is found in the Prefs drawer of the *FastTrak* boot floppy. It copies your current preference settings to the boot partition (or quick-boot floppy) where they are picked up when booting the computer. This must be done or your preference changes will not ''take.'' When using this utility on a quick-booting (not auto-booting) system, you must insert the quick-boot floppy.

MakeAutoBoot
MakeQuickBoot

These two utilities prep a system so that it will boot as quickly as possible.

MakeAutoBoot is for 1.3 systems to make them fully autoboot. *MakeQuickBoot* prepares a quick-boot floppy for 1.2 systems. See **Booting the System**, p. 19 for usage.

XetecFormat

This command, which can be found in the system drawer of the *FastTrak* boot floppy, is very similar to the normal system **format** command, but has some new options that are helpful for use with your hard drive. Its usage:

xetecformat DRIVE <partition> **NAME** <name>
[NOICONS][QUICK][FFS\NOFFS][NOQUERY]
[INHIBIT][HARDBLOCKS]

where <partition> is the device name of the partition to format, with <name> being the volume name given to it. As you can see, there are many options for this command. Let's take them one-by-one.

NOICONS: specifies that the partition should be formatted to be completely blank. If this option is not used, the reformatted partition will contain the trashcan and its icon.

QUICK: causes the partition to be reformatted by just emptying the directory and freeing all blocks. If omitted, every block in the partition will be read/write verified as well (takes quite a bit longer).

FFS\NOFFS: the system knows the handler used for each partition (as of the time it was installed) and will automatically use the correct format. You can specifically choose the format to use by including one of these options as follows: **FFS** for FastFileSystem or XetecFileSystem, and **NOFFS** for the standard handler.

NOQUERY: if this option is not used, you will be asked if you are sure you want to format the partition specified. You must type a "Y" or "N" and return. If you specify *NOQUERY*, the format will take place with no such question. Be very careful when using this option!

INHIBIT: using this option will cause the partition to be formatted, even if it has been write-protected using the AmigaDos **lock** command. Be careful!

HARDBLOCKS: this option is only intended for use by the *Partitions* program. If used, information about the partition being formatted will not be retrieved from DOS, but from the actual partition information stored on the drive. This is the only way to format a partition that is defined but not yet installed in the system.

BoardList

The *BoardList* command gives a detailed description of each auto-config expansion board attached to this computer **that is successfully configured in.**

DeviceList

The *DeviceList* command, located in the utilities drawer, must be used from CLI. Its purpose is to list all the devices currently installed in the operating system. Remember that each floppy disk unit and hard drive partition is also considered a device. Each device is listed along with all of its information.

Revisions

This utility must be used from CLI. It prints the revision number of the ROM in the hard drive interface and the revision of the harddisk.device driver. The SCSI ID of both is also printed (see **SCSI Networks**, p. 45).

SectorEdit

In the utilities drawer is a program, *SectorEdit*, which gives you direct access to specific blocks on your hard drive. **This can be very dangerous**, even for experienced users. To modify a file on the hard drive, use a file editor such as NewZap. Your hard drive is not the place to learn how to use and apply a sector editor. This documentation is intentionally brief, therefore, to discourage beginners from using it.

The sector editor works with one 512 byte sector at a time, which is displayed in two pages of 256 bytes (click the *Flip* gadget to alternate). The *Drive Unit* specifies which hard disk drive to work with, the *Cylinder*, *Head*, and *Sector* values specify the exact logical block desired on the drive. You can either enter the raw block number in the *Sector* gadget, or enter the correct number in all three. Click on the ends of the vertical arrow to advance to the next or previous sequential sector. The *Read* and *Write Sector* selections work as expected. Use the *Write To* box to rewrite to a different sector number (entered through the same *Sector, Head, Cylinder,* and *Drive Unit* gadgets). Point and click on the hexadecimal area or ASCII area to make alterations. Altered, unsaved data is shown in orange.

TouchAll

The *TouchAll* command does a ''touch'' to all file-system partitions to make their icons immediately visible from Workbench.

FORMATTING DRIVES

- *What is Low-Level Formatting?*
- *Using the* **SCSItools** *Utility*
  - *Editing Drive Parameters*
  - *Formatting the Drive*
  - *Adjusting the Interleave*
  - *Editing the Defect List*
  - *Defect Searching*
  - *Other Tools*
- **SCSItools** *Messages*

Note: This chapter deals with the details of low-level formatting. Since this can be complicated, skip right over it unless you have had a serious drive failure or need to prepare a new drive.

What is Low-Level Formatting?

Before a drive can be used to store data, its magnetic surfaces must be formatted. To be formatted means that certain marks and patterns are written onto the media that are used for synchronization and organization. This specific magnetic information can differ from one controller card to another, meaning that a drive must be formatted with the card that will normally operate it. (This obviously does not apply to "embedded" drives, which have their own SCSI controller built-in.)

Using the *SCSItools* Utility

On the *FastTrak* boot disk is a program called *SCSItools*. With the tools available in this program, you can take virtually any hard disk with one of several popular controller cards (if applicable), format it, test it, and prepare it for use in the *FastTrak* system.

To start *SCSItools*, either double-click on its icon (found in the Utilities drawer), or type *SCSItools* in CLI. After a few moments, the screen shown in fig. 6 should

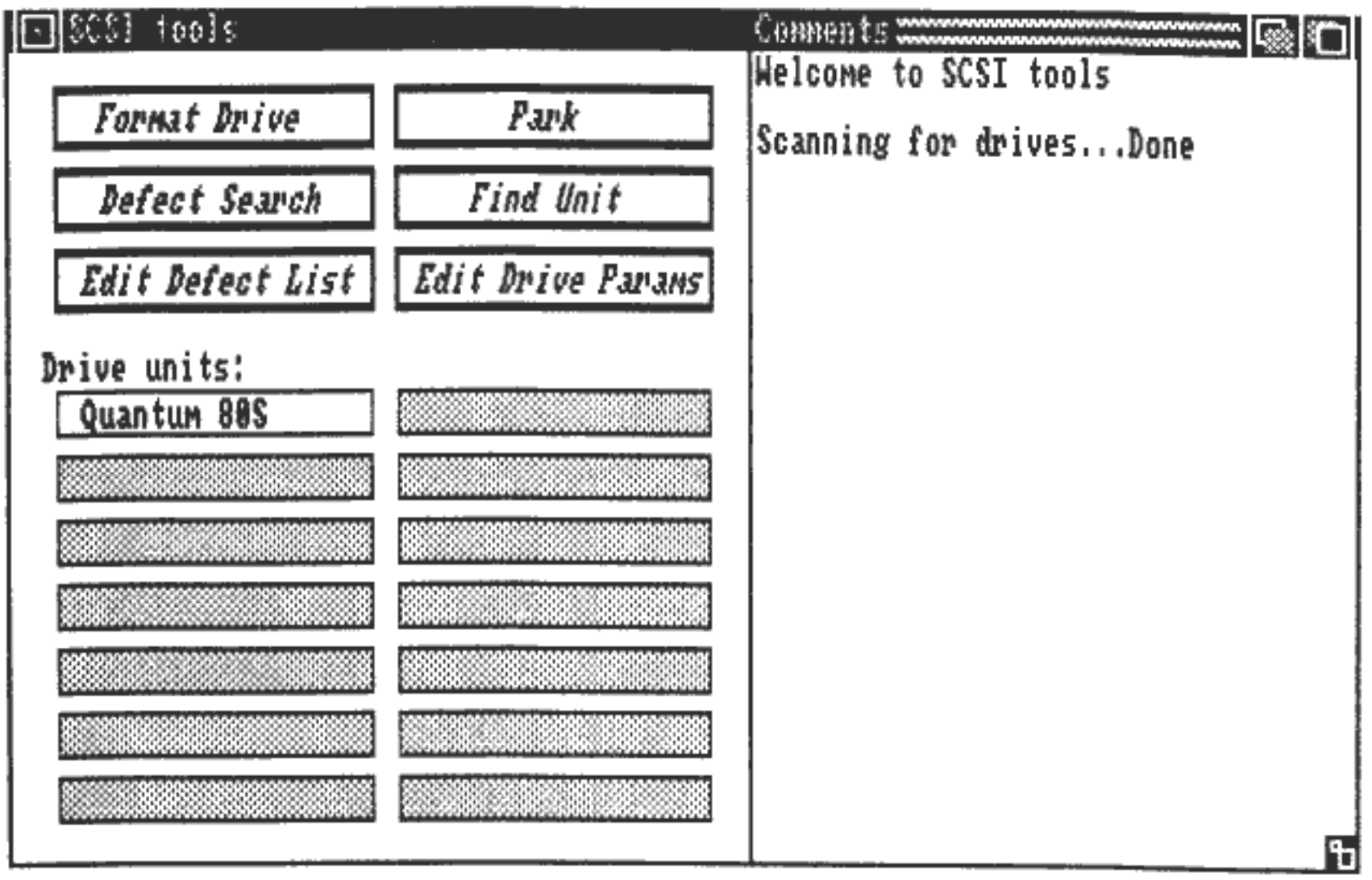

Fig. 6 – *SCSItools* Main Screen

appear. Let's touch on some points of interest. The small box in the extreme upper-left is the *close* gadget. Use it to exit the program. **Never** exit the program by shutting the computer off or rebooting! The program will save its final data as you exit. The six boxes in the upper-left are your commands. Clicking on these with the mouse allows you to perform the various functions of *SCSItools*. The area in the upper-right is a text window where the program can "talk" back to you, giving error messages and helpful info about what you're doing.

The 14 boxes marked **Drive units** represent the physical hard disk drives that can be attached to your *FastTrak* system. For every drive present, its name will appear in the corresponding box. The top pair corresponds to device address 0, the next pair to address 1, and so on thru address 6. For each pair, the left box represents LUN 0, the right one LUN 1. See *Unit Numbering* on p. 50.

Editing Drive Parameters

Before you can format a drive, you must make sure the parameters describing the drive and controller are correct. For a drive that is being formatted by *SCSItools* for the first time, default values will be picked, which you should promptly check and correct.

To select the drive to work with, click on its name box in the **Drive units** section. Its name should now be highlighted. If you have more than one drive connected, you will notice that only one can be highlighted at a time. Since each drive is a separate entity (with its own name, size, parameters, and partitions) you must work with one drive at a time.

Click on **Edit Drive Params** to begin editing the parameters for the selected drive. The window shown in fig. 7 should now appear. It contains all the user-changeable information for a drive. To edit any of the data, click on its box and use the **DEL**, **BACKSPACE**, and cursor keys. You can clear the data by holding and pressing **X**. Following is a description of each of the boxes in the edit window.

Bytes/Block: the size of data blocks this drive uses.
Cylinders: number of cylinders specified by drive manufacturer.
Heads: the number of magnetic surfaces or heads in the hard drive.
Sectors/Track: the number of data blocks that can be placed on a track for this hard drive. A track is the area of disk covered by one head in one revolution. Usually this is equal to the number of usable data sectors per track, except for the OMTI 3520 controller card which always keeps one sector per track for itself. For drives with a variable number of sectors per track, this number will be 0.
Interleave: correlates logical sectors to physical sectors on a track. Has a large impact on disk speed. See below for details.
Pre-Comp Cyl: cylinder number at which to start using write precompensation. See drive documentation. If not needed, set to 0.

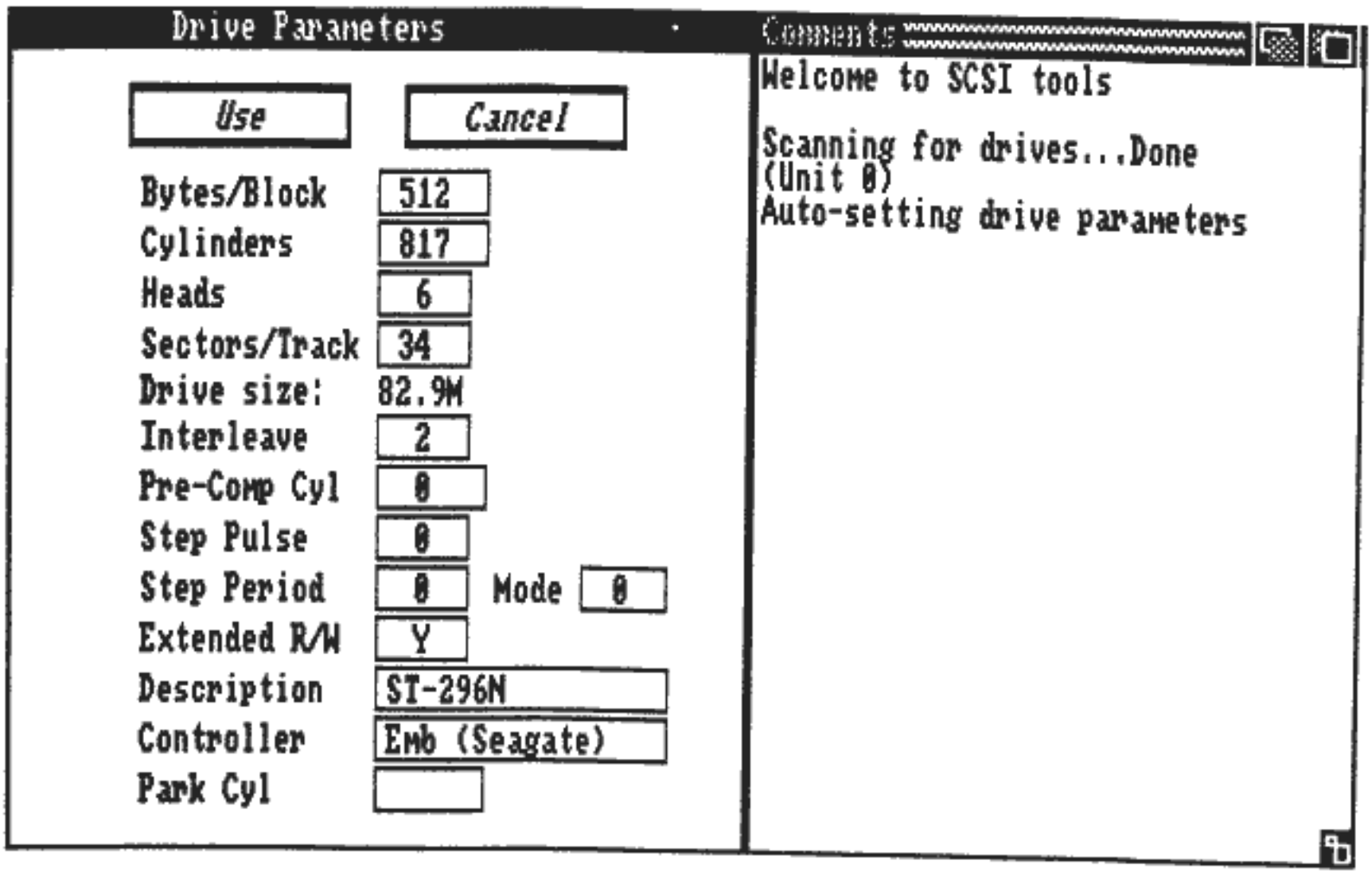

Fig. 7 – Drive Parameters Editing Screen

Step Pulse, Period, Mode: Very technical, controller dependent values. Do not change from factory values.

Extended R/W: flag for whether the hard drive supports the extended read/write commands ($28 and $2A). Higher transfer rates can be attained on drives that do. If this flag is set to "Y", *SCSItools* will test the drive and warn you if the drive doesn't like the extended commands. Leaving it at "Y" anyway can result in read/write errors later.

Description: a descriptive name of the controller/drive. This is the name that will appear in the "Drive units" boxes in *SCSItools* and *Partitions*.

Controller: a "multiple-choice" type value that changes every time you click on it. The *embedded* choices are for drives that need no controller card. If so, make sure the correct *embedded* version is used for the drive ("Sea" for Seagate, "Mini" for Miniscribe, or "Other").

Park Cyl: cylinder number where the heads can safely be parked. Depending on the controller selected, you may be prohibited from entering a value here (with some drives or controllers, the value is permanent). If using an OMTI 3520 controller, set this value to 4 less than the number of cylinders input.

To exit the editing window, you need to click either **Use** or **Cancel**. *Use* will save the current settings, where *Cancel* will not. Both return you to the main *SCSItools* screen.

Formatting the Drive

If you just edited the parameters for a newly-added drive, the next thing you should do is to format it. Any time you edit any parameters, you must reformat for them to take effect. Remember, this format **erases all data in every partition** on this drive. To start the format, click on the *Format Drive* box. A requester will appear to make sure you really want to proceed. If you answer positively the format will begin. Depending on the size and speed of the drive, this could take anywhere from a few seconds to five minutes. When completed, "Done" will appear in the text window.

> **ATTENTION**
>
> To guarantee the reliability of the hard drive, you should let the drive run for about 15 minutes so it can warm up before you do a low-level format. This is especially important if the drive is used in a cold environment or was just transported from one. If formatted cold, data errors may occur when the drive is warm.

Adjusting the Interleave

The interleave entry in the *SCSItools* drive parameters section can have a profound impact on disk speed. It specifies how the hard drive should arrange its sectors on the magnetic media (except for some drives, like Quantum, which ignore the interleave). Unfortunately, there's no simple formula to decide what value will give the fastest speed. Also note that the best interleave for another brand of host adaptor may not be the best interleave for your Xetec host adaptor.

For systems that came with a drive, the best interleave value has been set for you at the factory. If you have provided your own drive, however, you will have to experiment to find the best setting. Start with a value of 0 and move upward. For each setting, you must do a low level format (with *SCSItools*), format a partition (with *Partitions*), and run a speed test. Generally, you will see an increase in speed as you increase the interleave until suddenly the speed will drop way off. When this happens, you've gone "one too high." Back the interleave to the next lower value — this should give the best results.

Editing the Defect List

Every hard disk drive, no matter how new or well built, can have defects in its magnetic media. When drives are manufactured, they are tested rigorously. A list of the location of each defect found is included with the drive or attached

to its topside. (Drives with too many such defects are rejected.) These are called **factory defects**. If any other defects ever form over the life of the drive, they are called **user defects**.

In order for a drive to be reliable, the use of either type of defective area must be avoided. Hard drives accomplish this by a process called **reassigning**. For every defect on the usable part of the drive, the controller assigns a spare chunk of media elsewhere on the drive as its substitute. Once this is done, the drive won't attempt to use the defective area.

SCSItools takes care of all the details of this procedure for you; all you have to do is give it the list of defects to avoid. Click on *Edit Defect List* to bring up the defect list screen. Depending on the controller and drive, this screen may resemble fig. 8, 9, or 10. And depending on your drive, there may be already some defects listed.

First of all, you may see two separate defect lists (fig. 8 and 9) or just one (fig. 10). Embedded drives permanently remember the defects found at the factory, so no such list is kept in *SCSItools*.

Second, depending on the controller card used (if any), the headings in each defect list may vary. For example, the OMTI 3110 card only asks for the cylinder and head of each defect, whereas others need another parameter (block or BFI). Notice in the screen in fig. 8 that one list is made of Cyl/Hd/BFI while the other is Cyl/Hd/Blk. On drives where you must keep an external list of factory defects, they are specified by Cylinder, Head, and bytes from index (BFI). That's the way the list that comes with the drive is made out. For user defects (those discovered by you later), they are described by cylinder, head, and block (sector).

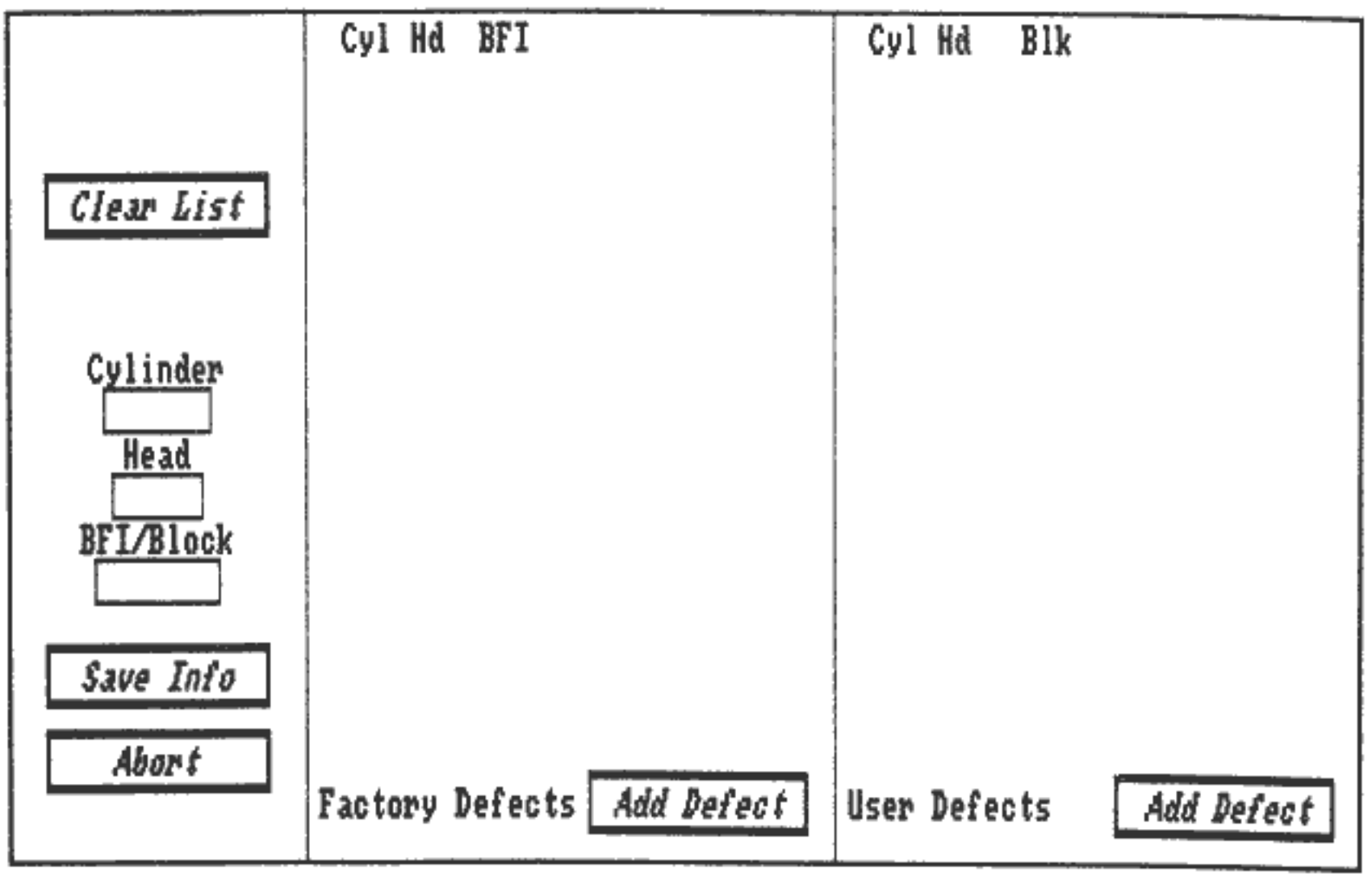

Fig. 8 – Defect List Screen (1)

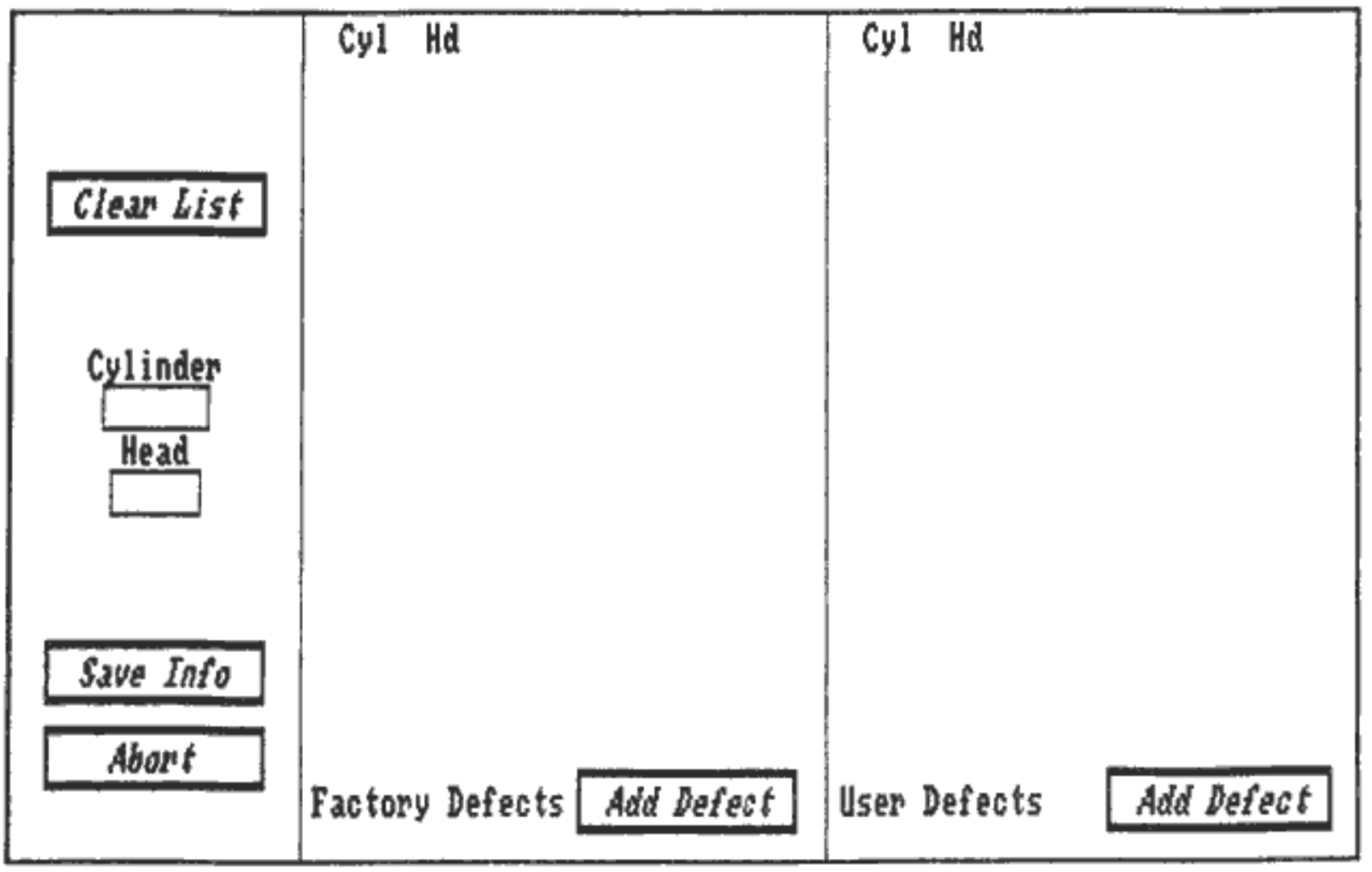

Fig. 9 – Defect List Screen (2)

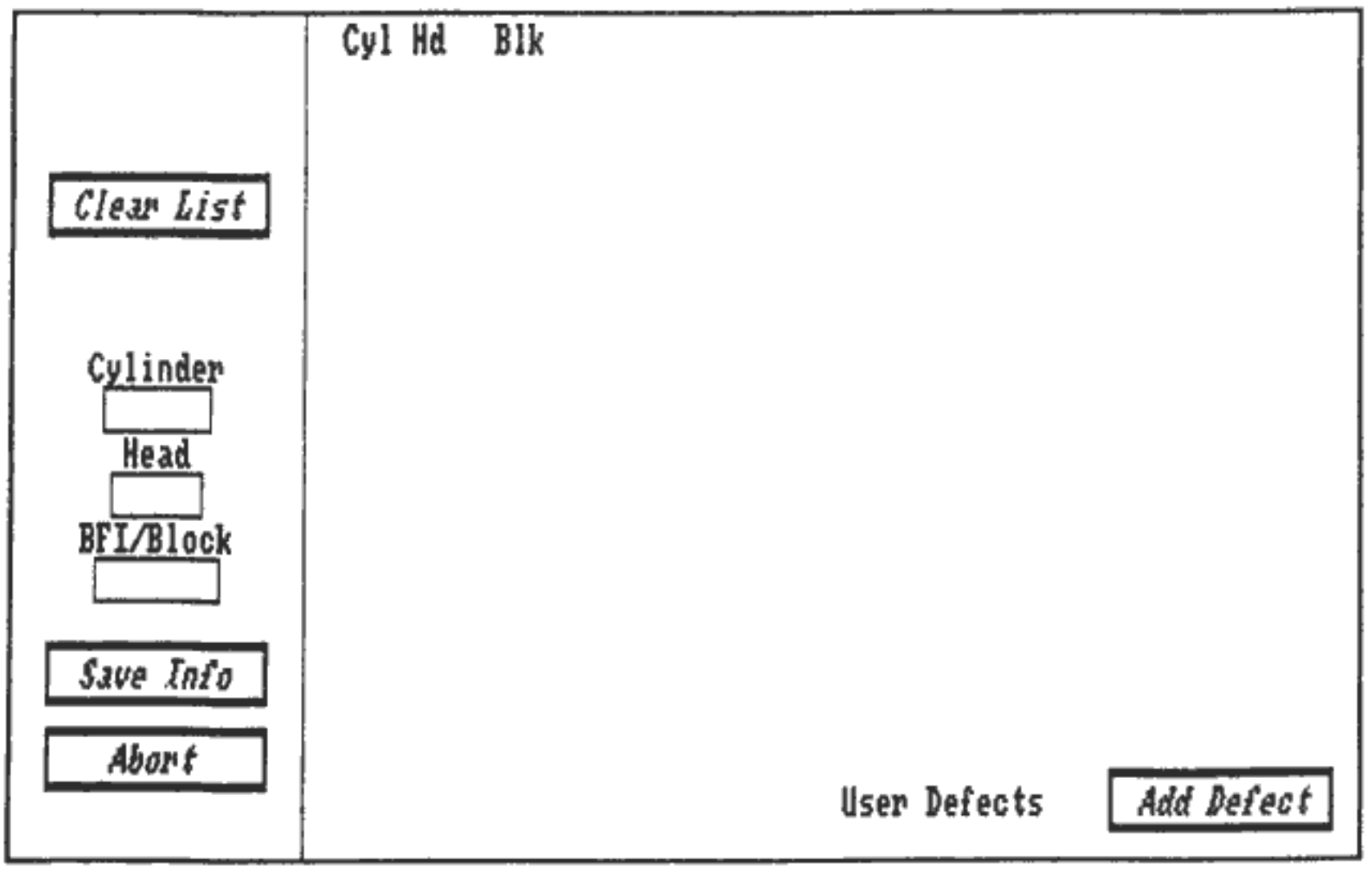

Fig. 10 – Defect List Screen (3)

Your *FastTrak* drive is shipped with the factory defect list already entered and the user list empty. The only time you will ever have to enter the factory list is for your own new drive, or if a drive's cylinder 0 ever becomes corrupt.

To actually add a defect description to one of the lists, enter its information in the boxes in the left margin marked *Cylinder*, *Head*, and *BFI/Block*. If the last box isn't there, then you don't need it. If it is there, enter the block number (for a user defect) or the BFI (for a factory defect). Once these boxes are filled in, click on **Add Defect** under the list you want to add it to. When you try to add a defect to the factory list, you will get a requester reminding you that this can never be erased (user defects can, as you will see later). If you are absolutely sure of its correctness, select the *Go ahead* choice. The new defect should appear at the bottom of the list.

If the screen flashes when you try to add a defect, that defect has already been entered, one of the values entered is illegal, or the list is full.

Note: *never* add any defects to the factory list that were not specified by the manufacturer of the drive. If you suspect other problem areas, list them in the user list — that's what it's there for.

If you ever want to erase the defects listed in the user list, you may do so by clicking on **Clear List**. A requester will appear, informing you that if you proceed with clearing the list, the drive will be reformatted (losing all data!). If you choose to proceed, the drive will be formatted, then the list will become empty. This action is permanent, even if you leave via *Abort*.

To exit the defect list screen, use **Abort** or **Save Info**. *Abort* will just return to the main screen, forgetting any list additions made. *Save Info* will permanently record the lists and will actually reassign all newly added defects. A requester will appear first, giving you a chance to change your mind.

Defect Searching

If you suspect additional defects have formed on a drive, you can test your theory by doing a **Defect Search**. This will test the writability of every data block on the drive. Although the test should leave the drive unaltered when done, it is wise to only do the test when you have the drive fully backed up. To run the test on the currently selected (highlighted) drive, click on *Defect Search*. A requester will appear to warn you that it may take a couple of hours to complete, and to give you a chance to chicken out. When the test is running, a counter in the text window will show the cylinder number currently being tested. If a bad block is found, its cylinder, head, and sector number will be displayed in the text window. You can abort the test at any time by clicking the *Defect Search* box once again.

It's possible for some drives to have so many defects that the list scrolls off the message window. For this reason, a "journal" of the latest defect search is kept in the RAM: disk under the filename "DiagSession". It can be read or printed for a complete record of the test results.

If any defects show up, copy down their location, and enter them in the drive's user defect list.

Other Tools

To test the "parkability" of a drive, click on the box marked *Park*. The selected drive will attempt to park to the cylinder input in the parameter editing screen (if applicable). A message will appear in the test window telling you if it succeeded or not.

The *Find Unit* command simply pulses the access light of the drive whose Drive Units box is currently selected. This is a good safeguard against confusion when, let's say, you have multiple drives and are about to format one of them. The first time you format the wrong drive, you will wish you had used this little device!

SCSItools Messages

The following are various messages that can occur in the text window of the *SCSItools* program:

Auto-setting drive parameters: an embedded drive is currently specified, so the drive has been examined and the parameters automatically set accordingly.

Doing low-level format...: the format is still in progress.

(Embedded drives dictate their own parameters): embedded SCSI drives have most of their parameters permanently fixed. This is just a reminder that your parameters are possibly being adjusted to match what the drive demands.

Error count: given at the end of a defect search test if any bad blocks were found. This is the total number.

error — Cyl:*n* Hd:*n* Blk:*n*: a bad block was found at this location during the defect search.

Error saving drive info!: an error occurred when trying to save the defect lists or drive parameters onto cylinder 0.

Fatal error, harddisk.device not found: *SCSItools* cannot even begin to operate because of the absense of this important system program. This file is only available when the system is booted from a disk that contains the Xetec icon and *harddisk.device* both in the *Expansion* drawer. This message also shows up if the host adaptor isn't configured into the system.

Improper controller specified: you specified that the drive has no controller (is embedded), but it did not respond like a typical embedded drive.

No errors detected: given at the end of a defect search test if no bad blocks were found.

No info on unit *n*: drive information created by *SCSItools* is absent on this drive. This is normal for a new drive.

Park failed: the drive would not park as it is currently setup. If this fails, so will the separate *Park* program.

Reassigning bad blocks: substitute blocks are being assigned to all defects in the defect lists for this drive.

Reassigning bad tracks: substitute tracks are being assigned to all defective tracks in the defect lists for this drive.

Sorry, I can't let you do that...: *SCSItools* won't let you test-park a drive which will stop spinning the platters. Of course the drive can be parked any other time (outside of *SCSItools*).

Testing unit *n*: the defect search test is running on the unit shown.

Test stopped: the defect search test was aborted by the user.

(Unit *n*): a new drive unit icon was highlighted; this is the unit number it is known as to the system.

Unit *n* isn't formatted: the unit shown is unformatted. You should edit its parameters and format it.

Unit parked at cylinder *n*: park procedure completed successfully. The separate *Park* program should work for this drive as well.

Writing factory defect list: the factory defect list is being passed to the controller card for it to use.

SCSI NETWORKS

- *Layout*
- *Drivers*
- *User Access*
- *Booting*

The *FastTrak* system interface and software now support SCSI bus arbitration. In plain English, this means that not only can multiple drives be connected on the bus, but so can multiple computers. This is an exciting prospect for the classroom or user-group who might be able to justify the cost of a multi-user hard drive.

Some limitations do exist. First of all, you can never have more than eight devices on one SCSI bus. Every drive counts as a device as does every computer. For example, you could have up to seven users and one drive, one user and seven drives, or anything between.

Layout

Setting up a network of SCSI drives and computers is in theory no different than a simple one-computer-one-drive system. The same SCSI philosophies must be applied: all devices must be arranged in a chain, and the two end devices must be the only ones with terminators installed. Here's a typical setup:

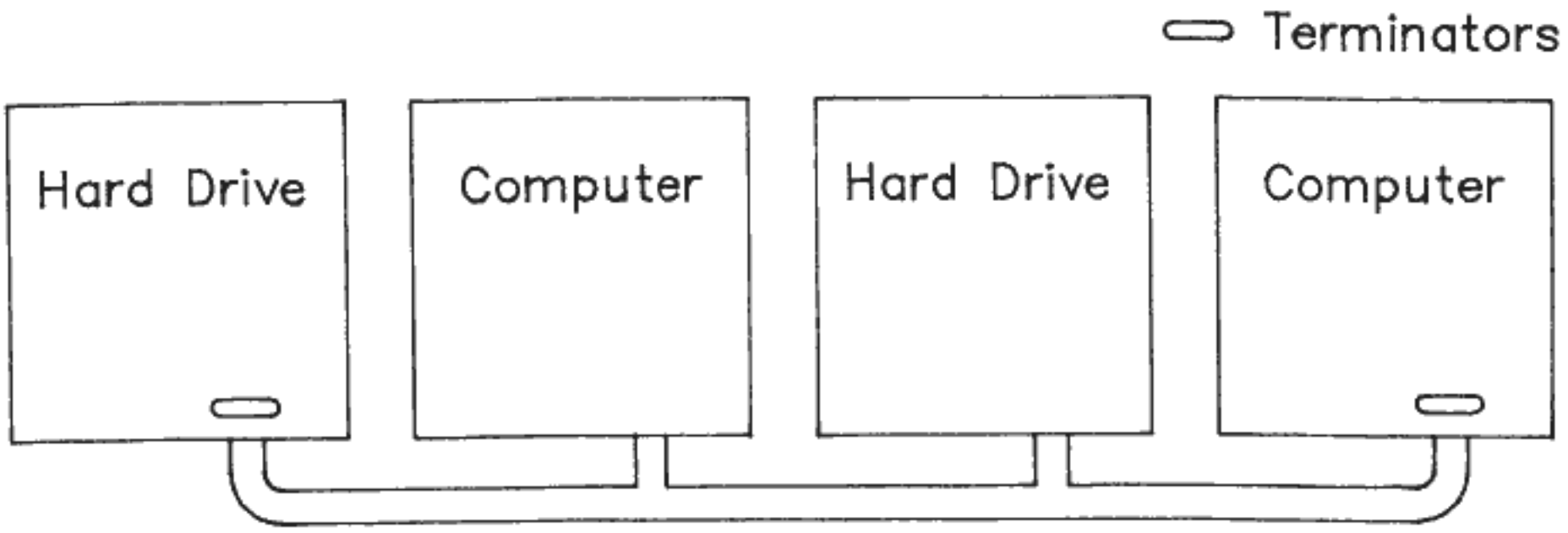

Fig. 11 — Sample SCSI Network

Drivers

Standard *FastTrak* systems come shipped as SCSI ID number 7. But just like each drive on the SCSI bus must have a different ID, so must each computer. Also, a computer and a drive cannot have the same ID number. Each successive system to be tied onto the same bus must be given a new ID number. The SCSI ID is selected by an "ID=n" tooltype in the "harddisk" icon in the boot partitions expansion drawer. A change to the ID tooltype is not immediately recognized, but is picked up the next time the computer is booted. Use the *Revisions* program to print the ID of the driver currently in use.

User Access

Using the *Partitions* utility, each partition can now be assigned to specific users with the new *Access* entry. If set to "all," any computer on the SCSI bus can use it. If set to a specific number, only a computer with that SCSI ID will be able to access it. An ideal way to use a system such as this is to partition the drive so that each user has a private partition (with its Access number set to the ID of its user), and a common partition that contains Workbench, utilities, etc.

File locking is not implemented, so care must be taken when writing to partitions that are shared among multiple systems (have an "all" status). Shared areas are useful mostly for common data areas which should normally be read-only anyway.

Booting

Only one computer on a network can autoboot, since the ID of the autoboot code is set to 7 for all host adaptors. ROMs with other ID's are available from Xetec, but a simpler solution is to quick-boot all the other systems. Refer to p. 22 for more info on quick-booting.

TECHNICAL INFORMATION

- *Unit Numbering*
- *Using the Hard Disk Driver*
- *Hardblocks*
- *Theory of Operation*

Unit Numbering

A deliberate numbering system is used by all *FastTrak* software to refer to each physical device on the SCSI bus. The circuitry on the hard drive interface uses this **unit number** to know which device should be accessed. Following are the valid unit numbers:

Unit	LUN	Address
0	0	0
1	0	1
2	0	2
3	0	3
4	0	4
5	0	5
6	0	6
10	1	0
11	1	1
12	1	2
13	1	3
14	1	4
15	1	5
16	1	6

The **LUN** entry stands for Logical Unit Number. This is for SCSI controller cards that can support more than one drive. These cards allow a LUN of 0 or 1 to distinguish between the two drives. Embedded SCSI drives and controller cards with only one drive attached will not respond to the "10+" unit number, because no LUN 1 exists for it. Knowledge of this unit numbering system will be helpful when initially setting up your drives and when using the included programs *Park*, *Unpark*, *DeviceList*, and *SectorEdit*.

Using the Hard Disk Driver

Programmers can directly access any SCSI device connected to the *FastTrak* host adaptor without going through AmigaDOS. Be very careful, one improper write can render a whole drive corrupt.

To open the *FastTrak* hard disk driver for use, from C:

error = OpenDevice("harddisk.device",unit,ior,0);

where ior is a pointer to an IOStdReq structure to be used to talk to the driver, and unit is the unit number of the device on the SCSI bus to work with. The function returns a zero if all went okay.

The following device commands are supported:

Type	Number
Read	2
Write	3
Direct access	28

The number listed is placed in the io__Command entry of the IOStdReq structure. The ''Direct access'' command conforms to the SCSIDISK standard which is emerging. Here's an example of a C function that uses this command.

```
#include <devices/scsidisk.h>

int SCSIdirect(size,unit,buffer,len,rw)
int  size,unit,rw;
UWORD *buffer;
ULONG len;
{
  int     status = -1, OpenFlag = 1;
  struct  MsgPort  *MyPort;
  struct  IOStdReq *Req;
  struct  SCSICmd  SCSIinfo;

  MyPort = (struct MsgPort *)CreatePort("ft__Direct",0);
  if (!MyPort) goto Aborter;
  Req = (struct IOStdReq *)CreateStdIO(MyPort);
  if (!Req) goto Aborter;
  OpenFlag = OpenDevice("harddisk.device",unit,Req,0);
  if (OpenFlag) goto Aborter;

  SCSIinfo.scsi__Data = buffer;
  SCSIinfo.scsi__Length = len;
  SCSIinfo.scsi__Command = (UBYTE *)&Packet;
  SCSIinfo.scsi__CmdLength = size;
  SCSIinfo.scsi__Flags = rw?SCSIF__READ:SCSIF__WRITE;

  Req->io__Command = 28;
  Req->io__Data = (APTR)&SCSIinfo;
  Req->io__Length = sizeof(struct SCSICmd);
  DoIO(Req);
  status = Req->io__Error;

Aborter:
  if (!OpenFlag) CloseDevice(Req);
  if (Req) DeleteStdIO(Req);
  if (MyPort) DeletePort(MyPort);
  return(status);
}
```

Notice the inclusion of the ''devices/scsidisk.h'' header file. Official developers should have this file on their distribution disks.

Hardblocks

All *FastTrak* software uses the new *hardblocks* data storage system. The format is very similar to that defined by the distribution include file ''devices/hardblocks.h'' but does differ in a few respects. The *hardblocks* system keeps all information about a drive and its partitions on the drive itself (on cylinder 0).

The older methods of mounting drives, such as editing entries into the mountlist and entering tooltypes in a special icon, have been eliminated by the *FastTrak* hardblocks system. The two utilities *SCSItools* and *Partitions* generate the hardblocks needed from the information you specify. This completely eliminates all the ''dirty work'' previously associated with hard disk setup and maintenance.

When installing drives, the *FastTrak* system software looks for hardblocks. It uses the data in the hardblocks to know if and how to install the drives and partitions into the operating system. This makes a beautifully clean system: if a drive is disconnected, the hardblocks for it disappear too, so the system doesn't try to install a drive that doesn't exist. In fact, you can connect, disconnect, and change the drives unit numbers (via the jumpers) at will, and each will still be installed as expected.

Manually modifying a drive's hardblocks is not recommended, as you could confuse one of the system programs that access them.

Theory of Operation

Figure 12 gives a basic overview of the software ''chain of command'' between your programs that access the hard drive and the actual hardware. Most programs give requests for data read/writes to AmigaDOS which then dispatches it to the specific handler assigned to that partition. This is the handler talked about in the section describing the *Partitions* program. The handler directly talks to *harddisk.device*, the assembly language driver that actually controls the hardware in the hard drive interface. It turns all requests given to it into electrical signals passed along the SCSI bus. This whole chain works in either direction, depending on the type of transfer requested (read or write).

Notice the level at which *SCSItools* and *Partitions* operate; they talk directly to the hardware driver. Until a partition is installed into the operating system, it is invisible above the dotted line. This graphically explains why you can talk to a drive with *SCSItools* or *Partitions* any time, but can't use the drive unless it's installed.

For special applications not needing file-oriented storage, but where raw storage at maximum speed is a necessity, it is possible to write programs that bypass the DOS and handlers, and just talk to *harddisk.device* directly. Programmers with this in mind should use the "Direct access" I/O command type mentioned above in *Using the Hard Disk Driver*.

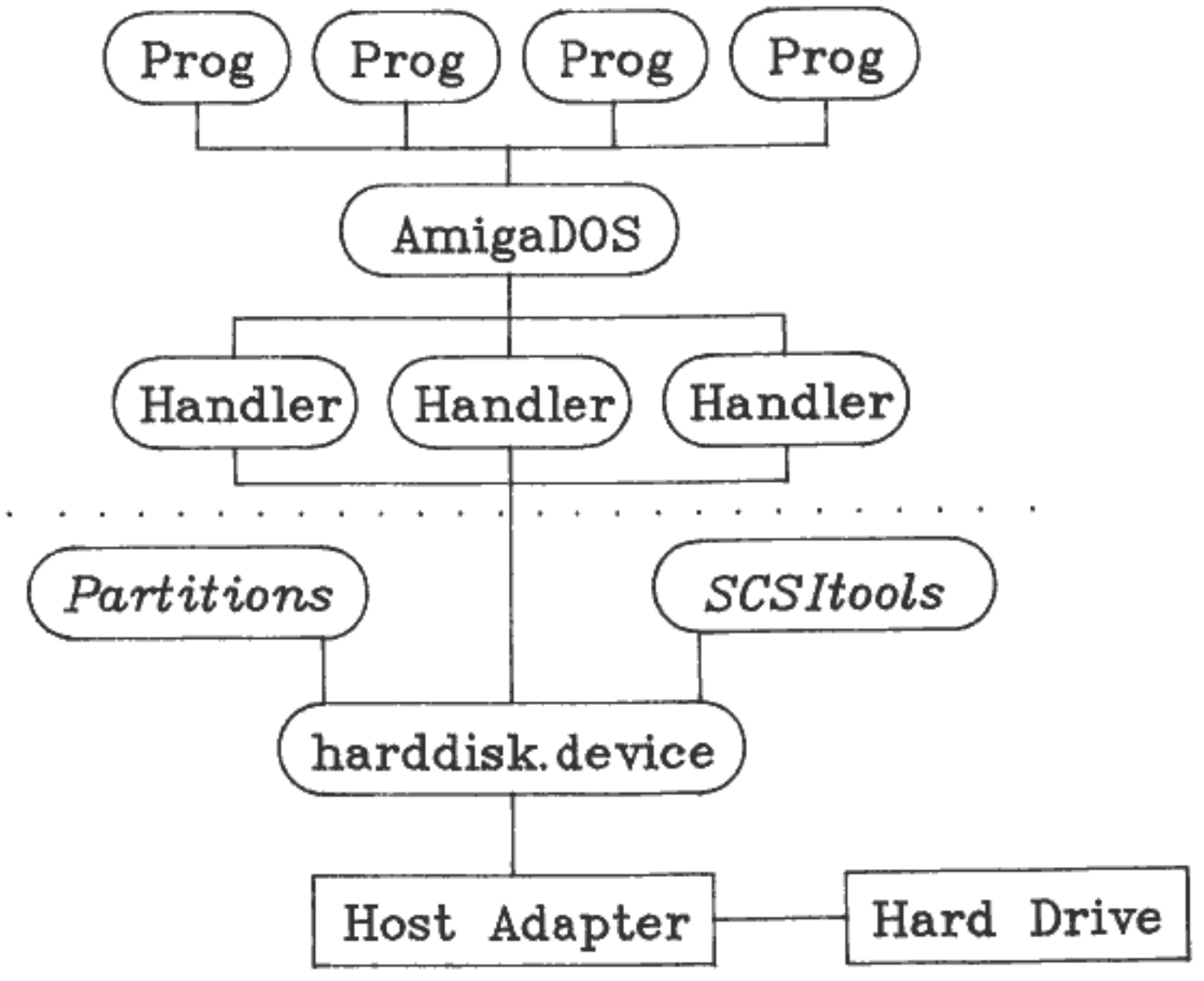

Fig. 12 — *FastTrak* Software Interface

TROUBLESHOOTING GUIDE

Symptom	Possible Cause
Drive won't turn on	Power cord not seated fully in back of drive Bad outlet Blown fuse in back of drive
Computer won't boot	Wrong operating system jumper setting in host adaptor Wrong power-up sequence Host adaptor plugged in improperly
System locks up when accessing hard drive	Improper termination on SCSI bus Data cable not screwed on at both ends
Partitions not usable from DOS	Computer must be rebooted after partition defined, before it's usable
"Not a DOS disk" error or "Error Validating Disk"	Trying to install or use an unformatted partition. Drive reformatted with wrong format for handler (FFS or NOFFS).
"Volume xxx has a read/write error"	Using a hard disk that has been parked A defect was not reassigned properly. A new defect has developed that needs to be reassigned
Drive makes odd sounds or makes sounds at inappropriate times	Different hard drives make different sounds, some even when no accessing is being done. A grinding noise is ungood, though.

CUSTOMER SERVICE

If you have a problem that is not covered in this manual, you may obtain help by calling customer service at (913) 827-0685 between 10 a.m. and 4:30 p.m. CST. **We cannot help those who have not read this manual.** Before you call, have some information ready: find the revision of your *FastTrak* floppy (see its icon or disk name); also write down the drive and partition parameters for the problem drive (use *SCSItools* and *Partitions* to view them). This information will likely be vital in our solving your problem.

If you discover a bug or would like to make recommendations, use the bug report form on p. 58. Please leave our service lines open for those who need answers. Bug report forms are also more quickly and easily routed to those who need to see them.

Xetec also has a bulletin board in operation to provide support for all *FastTrak* owners. The number is (913) 827-1974.

EQUIPMENT RETURN POLICY

Do not return any system or part thereof unless you have first called and received authorization. Any hardware to be returned will require an RMA (return materials authorization). Any hardware without an RMA number clearly marked on the outside of the box will be refused. Call (913) 827-0685 to obtain an RMA.

When shipping a drive, remember to use the *Park* command (see p. 32) and to use the original carton. Include a letter describing the problem and proof of purchase, if under warranty.

The drive's warranty will only be honored if the registration card has been completed and mailed in.

UPDATES AND ERRATA

Obtaining Updates

To be notified of important system updates, you must be registered with Xetec. To do this, you should complete your registration card and mail it in. Certain enhancements and/or bug fixes are bound to follow. Depending on the nature of the update, a minimal charge may be required.

README File

Any up-to-the minute changes that don't make it into the manual will appear in this text file, located on the *FastTrak* boot floppy. It may also contain new features, hints, and caveats. To read the file, either use

type readme

from CLI, or from Workbench double-click on its icon. On the hard drive should be a README file explaining how to use some of the demos included on the drive.

To obtain a hardcopy of any of these files, type this from CLI

type >prt: filename

Bug Report Form

Unfortunately, even the most refined programs can contain bugs. The most common bug is the one that occurs with a specific combination of hardware and usage that was never foreseen or tested. We have made every effort to eliminate all such bugs. If you discover what you think is a bug, please report it as soon as possible on a copy of the following form. We truly wish to make the *FastTrak* system error-free, and will give every formal bug report close and careful consideration.

If you find multiple problems, please submit each on a separate copy of the form. A copy of this form is in the file *BugForm* which you may print and use (type >prt: BugForm).

FastTrak Bug Report Form

Date:
Name:
Company:
Phone: ( ) -

Bug type: ☐ Hardware
☐ Software
☐ Documentation
☐ Other

Revisions: ____ Kickstart ROM or disk
____ Workbench
____ FastTrak boot floppy
____ Program in question (if software)

Computer type _____
Memory _____
Other equipment _____

Bug description:

Bug generation procedure:

GLOSSARY

Archive bit: a bit kept for every DOS file that signifies whether it is currently backed-up on floppy. This bit is visible using the DOS *List* command.

Auto-boot: the process of initializing or booting the computer to operation using a hard drive as the medium, rather than a floppy.

Block: the smallest organized chunk of data on a hard drive. Usually equal to 512 bytes. Also called a *Sector*.

Boot partition: a very small partition that is only used during the auto-boot phase.

Click: pointing at an object with the mouse, then pressing and releasing the left mouse button.

Controller card: an adaptor needed with some hard drives to allow them to communicate on a SCSI bus.

Cylinder: a group of tracks, one for each head.

Defects: defective or marginal portions of a hard drive that must not be used.

Device address: the unique number given to each device connected to the *FastTrak* SCSI bus chain.

DMA*x*: innovative hardware mechanism on all Xetec host adaptors which give them their speed.

DOS: the software mechanism that takes care of input/output transactions. It intercedes between application programs and the handlers for each mass-storage device.

Double-click: pointing at an object with the mouse, then pressing and releasing the left mouse button twice quickly.

Drive unit: each individual hard drive or peripheral connected to the *FastTrak* SCSI bus chain.

Embedded drive: a hard drive which has a controller card embedded within it. These drives can usually be connected directly to a SCSI bus system.

Handler: the software mechanism that connects DOS to the software driver for each storage device. The handler deals with data in a file format.

Hardblocks: special data concerning each hard drive (stored in cylinder 0).

Heads: number of magnetic read/write heads in a hard drive — same as surfaces

Host adaptor: the hardware interface that allows your computer to communicate with SCSI devices.

Incremental backups: copying only new or modified files, to save the time of reduntantly backing-up files that haven't been modified.

Lock-up: a serious problem with a computer that causes it to behave abnormally, or even to stop dead in its tracks.

LUN: logical unit number — designates the specific drive desired when using a controller card than can handle two hard drives.

Partitions: sub-divisions of the storage of a hard drive, each division treated as a separate entity.

Reassigning: to assign a substitute area of storage for a defect.

Requester: a window which appears while using a program to double-check before doing something drastic.

Script file: a type of text file, usually located in the *s* directory, which contains command lines. Typing **execute *scriptfile***, each line in the file is performed just as if you had typed it from the keyboard.

SCSI: the data protocol used in the *FastTrak* system to talk to the hard drives (stands for ''Small Computer Systems Interface'').

Sector: synonym for *Block*.

Surface: the area of magnetic media in a hard drive, accessible by one head. The number of surfaces and heads are always equal.

Terminators: Resistor pack(s) installed in hard drives or controller cards to comply with the design of the SCSI bus.

Track: the magnetic area in a hard drive covered by one head in one disk revolution.

Unit number: the number used among *FastTrak* software to specify the exact hard drive or SCSI peripheral to be used.

Zaphod Beeblebrox: worst dressed sentient being in the known universe seven times running.

INDEX